# Division of space by Voronoi graphs, application to the models of porous membranes,

A thesis submitted to the

University of Manchester Institute of Science and Technlogy

For the degree of Doctor of Philosophy

**Kittisak Nui Tiyapan**

BEng, BSc, BEng, MSc

Department of Chemical Engineering

$23^{th}$ December, 2002

**Declaration**

The work that § E.1 in this thesis is based on has been submitted in support of an application for the degree of B.Eng. in Mineral Engineering at the Chulalongkorn University, Bangkok, Thailand, in the year 1991 AD.

Kit Tiyapan

To my father and my mother Niwat and Somjit Tiyapan,
and my chemistry and sword teacher Siddhiponr Songsataya.
To Emeritus Professor David J. Bell
and Emeritus Professor Graham Arthur Davies,
both of whom have started this project,
one of whom has powered it.
To these people, then, is my dedication of this thesis,
but the thesis may dedicate itself to whomever it wishes.

## Table of contents

b. List of figures ..... 6
c. List of tables ..... 9
d. List of theorems ..... 10
e. List of algorithms ..... 10
f. List of definitions ..... 11
g. List of assumptions ..... 11
1. Introduction ..... 13
1.1. Mathematics ..... 17
1.2. Geometry ..... 18
1.3. Physics ..... 27
1.4. Cosmological structure ..... 30
1.5. Filtration ..... 31
1.6. Statistics ..... 34
1.7. Poisson process ..... 35
1.8. Phase transition ..... 36
1.9. Random processes ..... 36
1.10. Structures in nature ..... 42
1.11. Computational geometry ..... 44
1.12. Geometric algorithms ..... 50
2. Division of space ..... 51
2.1. Stereographic projection ..... 53
2.2. Covering lattices ..... 56
2.3. Viscous fingers ..... 58
2.4. Crystals, quasicrystals, and polycrystals ..... 59
2.5. Random lines ..... 63
2.6. Convex hull ..... 63
2.7. From convex hull to the Voronoi tessellation ..... 65
2.8. Tilings and patterns ..... 71
3. Voronoi tessellation ..... 75
3.1. Quadratic equations ..... 78
3.2. Quadratic forms ..... 78
3.3. Voronoi algorithms ..... 79
3.4. Voronoi statistics in literature ..... 79
3.5. Voronoi statistics ..... 81
3.6. Voronoi section ..... 84
3.7. Number of vertices and edges ..... 85
3.8. Faces in different dimensions ..... 101
3.9. Beam intersection study ..... 107
3.10. Voronoi of Voronoi ..... 111
3.11. Transformations of Voronoi ..... 113
3.12. Compressed Voronoi ..... 113
3.13. Voronoi tessellation in higher dimensions ..... 123
4. Percolation ..... 125
4.1. S-curves and the percolative phenomena ..... 131
4.2. Voronoi percolation in two dimensions ..... 132
4.3. Voronoi percolation in three dimensions ..... 134
4.4. Percolation of 2-dimensional Voronoi sections ..... 136
4.5. Network percolation ..... 138
4.6. Percolation statistics in literature ..... 138
4.7. Percolation of $n$-gons in continuum ..... 140
4.8. Polygon percolation threshold ..... 143
4.9. 2-homohedral tilings ..... 148
4.10. Cosmology ..... 162
4.11. Philosophy ..... 163

4.12. Enlightenment ..... 167
4.13. Forest fire fighting ..... 168
4.14. Fractals and percolation ..... 172
5. Porous media ..... 173
5.1. Medical science ..... 173
5.2. Zeolites ..... 176
5.3. Crystalisation ..... 176
5.4. Fluid flow within networks ..... 177
5.5. Material science ..... 178
5.6. Forces between particles ..... 181
5.7. Arbitrarily shaped particles ..... 181
5.8. Non Poisson number distributions of particles ..... 183
6. Filtering membranes ..... 185
6.1. Separation processes ..... 195
6.2. Dead-end filtration ..... 195
6.3. The centre of gravity ..... 195
6.4. Molecular dynamics ..... 199
6.5. Problem definition and algorithms ..... 201
6.6. Simplified algorithm for filtration ..... 203
6.7. Filtering problem when physical blockage is prominent ..... 204
6.8. Percolative filtering with very small particles ..... 204
6.9. Percolation within percolation ..... 207
6.10. The first part, suspended particles ..... 209
6.11. The second part, flow through the cells ..... 214
7. Percolation in traffic modelling ..... 219
7.1. Traffic networks ..... 223
7.2. Percolation of road networks ..... 223
7.3. Theory of graphs and its applications ..... 229
8. Percolation theory as economic models ..... 230
8.1. Object-location ..... 230
8.2. Economic modelling ..... 235
9. Addenda ..... 238
9.1. Sociology ..... 238
9.2. Control systems ..... 240
9.3. TEXnicalities ..... 247
10. Conclusion ..... 248
A. Programs ..... 249
A.1. Object location ..... 249
A.2. Network percolation, two dimensions ..... 250
A.3. Network percolation, three dimensions ..... 253
A.4. Network percolation, 2–d section ..... 258
A.5. Continuum percolation of $n$-gons ..... 261
A.6. Tilings ..... 263
A.7. Covering lattices ..... 275
A.8. Covering contour ..... 276
A.9. Number of vertices ..... 277
A.10. Vertices per cell and cell ratio ..... 277
A.11. TEX's macros ..... 278
A.12. Language macros for TEX ..... 286
A.13. Degenerate test and other programs ..... 299
A.14. Face statistics in $n$ dimensions ..... 300
A.15. Beam intersection program ..... 302
A.16. Number of vertices in high dimensions ..... 302
A.17. Example batch program, simulation and data extraction ..... 303
A.18. Voronoi operator of various order ..... 303
A.19. Voronoi data structure for filtering membrane study ..... 304
A.20. Centroid process on 2-d VT ..... 307
A.21. Centroid process on 3-d VT ..... 308

A.22. Defining irregular objects ..... 311
A.23. Investigating the distribution observed from within the network ..... 314
A.24. Miscellaneous functions ..... 315
A.25. Percolated traffic networks ..... 317
A.26. Volume, surface area, cell- and face perimeters ..... 324
A.27. Volume in higher dimensions ..... 329
A.28. Regular lattices in three dimensions ..... 330
A.29. Effects of channelling ..... 334
A.30. Stereographic projection ..... 340
B. Terminology ..... 342
B.1. Abbreviation ..... 346
C. Resources ..... 347
C.1. Biographies ..... 347
C.2. Computation and softwares ..... 349
C.3. Internet resources ..... 351
D. Bibliography ..... 352
D.1. My writings, Kittisak Nui Tiyapan ..... 357
E. Essays and relevant publications ..... 359
E.1. Antimonytrioxide extraction from ore by hydrometallurgy ..... 359
E.2. Cyberspace ..... 366
E.3. The End of Stars ..... 367
E.4. Self-tuning Extremum Control ..... 369
E.5. The story of Andromedra ..... 379
E.6. The names of the notes ..... 380
E.7. To be unkempt ..... 382
E.8. Critical probability of 2-d tessellation ..... 382
E.9. Some properties of stochastic optimal control ..... 385
E.10. Fractals in traffic control ..... 387
E.11. Distributed parameter systems ..... 387
E.12. Vision robots ..... 388
E.13. Singular perturbation ..... 389
E.14. Object-location using Extremum Control ..... 390
E.15. On pragmatists and idealists ..... 393
E.16. The Morris Worm ..... 396
E.17. RLS algorithm for object-location problem ..... 398
E.18. Modelling the economics ..... 400
E.19. Money flow in networks ..... 401
E.20. Modelling of traffic congestion ..... 404
E.21. Critical probability in traffic modelling and control ..... 413
E.22. Variable structure control and singular perturbation ..... 415
E.23. Projects on translation ..... 417
E.24. Abstracts of books I wrote ..... 420
F. Translation ..... 423
F.1. G. L. Dirichlet, 1848 ..... 423
F.2. G. F. Voronoi, 1908 (I) ..... 430
F.3. G. F. Voronoi, 1908 (II) ..... 471
F.4. G. F. Voronoi, 1909 ..... 511
G. Index ..... 561

## List of figures

1.1. Kissing number versus dimension ..... 13
1.2. Truncated tetrahedron and octahemioctahedron ..... 17
1.3. Tetrahemihexahedron and truncated octahedron ..... 18
1.4. Truncated cube and rhombicuboctahedron ..... 18
1.5. Pentagrammic polyhedra ..... 18
1.6. Temperature of the universe ..... 24
1.7. Temperature of the universe according to the Big Bang theory ..... 24
1.8. Voronoi graph with bridge-shaped beta distribution ..... 32

1.9. Voronoi graph with J-shaped beta distribution....32
1.10. Voronoi graph with U-shaped Beta distribution....32
1.11. Voronoi graphs of various random p.d.f.'s....33
1.12. Point process $r(\theta)$, where both $r$ and $\theta$ are independent random generators....35
1.13. Point process $x, y$ in a circular domain, where $x$ and $y$ are independent random generators 36
1.14. Point process $x, y$ in a square domain, where $x$ and $y$ are independent random generators . 36
1.15. Additional generators for the incremental method....45
2.1. Stereographic projection of a cube....48
2.2. Rotated cube....48
2.3. Developmental stages of programs for the stereographic projection....50
2.4. Covering lattice, stone pavement, or galaxies in the forming....52
2.5. The area of multiply covered cells....53
2.6. Face number of non isometric crystal forms....56
2.7. The 71 Voronoi cells in an aggregation....60
2.8. Four groups of tiles....67
2.9. Percolation of networks related to the square lattice....68
2.10. Percolation of networks related to the honeycomb lattice....69
2.11. Percolation of networks related to the basket pattern, kagome lattice....69
3.1. Rooms with doors puzzle....70
3.2. Departure and arival rooms....71
3.3. Voronoi from degenerative data....71
3.4. Shifted hexagonal lattices....71
3.5. Hexagonal, Kagome, and the covering of the latter....72
3.6. The next covering lattice....72
3.7. Voronoi graphs of covering lattices....72
3.8. The minimum of the average normalised edge length....77
3.9. The maximum of the average normalised edge length....77
3.10. The mean of the average normalised edge length....77
3.11. Mean, maximum, and mean of the cell-average normalised edge length....78
3.12. The variance of the expected values of the normalised length of edges of a cell....78
3.13. Number of vertices per cell....79
3.14. Ratio of cells in the original domain....79
3.15. Sectioning by a plane....79
3.16. Abundance of elements....81
3.17. The electrical resistivity and conductivity of elements....81
3.18. Space covered by Voronoi....82
3.19. Number of vertices versus number of cells....83
3.20. Number of vertices versus number of cells....83
3.21. The CPU time in seconds....83
3.22. The CPU time in seconds....83
3.23. Characteristic length versus the number of cells....83
3.24. Characteristic length versus the number of cells....83
3.25. Characteristic area versus the number of cells....83
3.26. Characteristic area versus the number of cells....83
3.27. Distribution of the number of edges per face....84
3.28. Statistics of a 231 cells Voronoi structure....87
3.29. The distributions of the number of faces and edges per cell....87
3.30. $v_n/c_n$ in Voronoi networks of various dimensions....90
3.31. Ratios between the number of vertices and the number of cells in Voronoi networks of various dimensions....90
3.32. Cpu time in creating the Voronoi networks....90
3.33. Cpu time in creating the Voronoi networks for Figure 3.33....90
3.34. Distribution of vertices in high dimensions....90
3.35. Counting valences....96
3.36. Distribution of $v_c$ in two dimensions....96
3.37. Distribution of $v_c$ in each simulation on 2-d Voronoi....102
3.38. Distribution of vertices in 3-d Voronoi....98
3.39. Distribution of vertices in 4-d Voronoi, 100 nuclei....100

3.40. Distribution of vertices in 4-d Voronoi, 300 nuclei points....101
3.41. Distribution of vertices in 5-d Voronoi, 200 nuclei points....102
3.42. Intersection by a line....102
3.43. Line intersection, Rayleigh distribution....106
3.44. Line intersection in Rayleigh distributed Voronoi....106
3.45. Two Voronoi operators applied in succession....107
3.46. Applying the Voronoi operator twice in succession....107
3.47. The Voronoi operator applied six times....107
3.48. The number of cells and vertices when recursively applying the Voronoi operator....108
3.49. Recursively applying the Voronoi operators of various orders....108
3.50. The effect of $z$-axis compression....108
3.51. Mean and standard deviation of surface area of cells under compression....108
3.52. Some hidden characteristics revealed....109
3.53. Volume of compressed Voronoi cells....110
3.54. Cell perimeter....111
3.55. Face perimeter....112
3.56. The cell perimeter and the face perimeter....113
3.57. Area, volume and perimeter; past results....115
3.58. Statistics of face areas under compression....116
3.59. Number of generators required for ten inner cells....119
3.60. Mean and standard deviation of the ratio Voronoi per defining cube....120
3.61. The ratio $V_p/V_h$, dimension from 1 to 6....120
4.1. predicted value of the heat capacity of monoatomic solids according to Debye....127
4.2. Section by the plane (0.01, 0.5, 0.5, -0.5)....132
4.3. Section by the plane (0.5, $-0.5$, 0.5, 0.01)....132
4.4. Section by the plane ($-0.7$, $-0.3$, 1, 0.01)....132
4.5. Regular polygon....135
4.6. Star....135
4.7. Tilting....136
4.8. Aggressiveness....137
4.9. $n$-gons....138
4.10. Finding the threshold area ratio of pantagons....139
4.11. Percolation of $n$-gons....141
4.12. Thirty-nine 2-homohedral tilings....143
4.13. Sample trend of the biggest, the second biggest, and the smallest cluster....148
4.14. Basic units of the thirty-nine 2-homohedral tilings....149
4.15. Range of observatory towers....165
4.16. Tortuous boundaries....165
4.17. Porcupine diatram superimposed on a map....166
5.1. Schematic diagram of a separator....174
5.2. Coulomb force *vs* distance....176
5.3. Coulomb force *vs* distance in log-log scale....176
5.4. *Kit* the existential object....177
5.5. Distributions by weight and by number....178
5.6. Spherical object on a sieve....180
6.1. Centroid of the Voronoi tessellation....191
6.2. Evolution of nuclei towards centroids....192
6.3. Evolution of nuclei towards centroids in three dimensions....193
6.4. Change in size distribution as a result of a nuclei adjustment process....194
6.5. Lennard-Jones potential....196
6.6. Force corresponding to the Lennard-Jones potential....196
6.7. Force balance with one particle fixed....201
6.8. The packing density calculation of the closest-packed densities....202
6.9. Percolation within percolation....203
6.10. Percolation of the cubic close-packed lattice....207
6.11. The eight areas defined by the eight types of unit blocks....209
6.12. Intersection of convex hull faces and the horizontal plane....211
6.13. Correcting the effect of rounding or discretisation....211

6.14. After increasing the precision ..... 211
6.15. `Spying` the neighbours ..... 212
6.16. Sections of bonds and cells ..... 213
7.1. Traffic status, $p_c > 0.5$ ..... 215
7.2. Traffic status, $p_c < 0.5$ ..... 216
7.3. Traffic control algorithm ..... 216
7.4. Proposed traffic control in real time. ..... 216
7.5. Manchester road networks ..... 219
7.6. Plots of the largest clusters, Manchester ..... 220
7.7. Road networks of Amsterdam ..... 220
7.8. Plots of the largest clusters, Amsterdam ..... 221
7.9. Road networks of Brussel ..... 222
7.10. Plots of the largest clusters, Brussel ..... 222
7.11. Road networks of Freiburg ..... 223
7.12. Plots of the largest clusters, Freiburg ..... 223
8.1. Square object located by a square mask. ..... 226
8.2. A snapshot of of the object and eight others of the mask ..... 226
8.3. The hill in an object location problem. ..... 228
8.4. Contour of a hill ..... 228
8.5. RLS procedure with random search ..... 229
8.6. RLS procedure with random search and back-stepping. ..... 229
8.7. Economical models. ..... 232
9.1. Phase plane of Equations 9.34 and 9.35 with no input ..... 236
9.2. Phase plane of Equations 9.34 and 9.35 with signum function in the input ..... 236
9.3. The eigenvalues of the dc motor ..... 238
9.4. The eigenvalues of the dc motor as a function of $\varepsilon$. ..... 238
9.5. Dynamics of the mechanical and the electrical angles ..... 238
9.6. State plane of the system $\varepsilon \mathrm{d}^2x/\mathrm{d}t^2 + \mathrm{d}x/\mathrm{d}t = u$ with $0 < \varepsilon < 1$. ..... 240
9.7. State plane of $\varepsilon \mathrm{d}^2x/\mathrm{d}t^2 + \mathrm{d}x/\mathrm{d}t = \mathrm{sgn}(x_1 + x_2)$ when $\varepsilon \geq 1$. ..... 241
10.1. Example of a relationship between a performance index and two inputs. ..... 364

## List of tables

1.1. Three types of function ..... 17
1.2. Perimeter per unit area of $n$-gons ..... 20
1.3. Some important properties of regular solids ..... 20
1.4. Surface area per volume of regular solids ..... 21
1.5. The number of vertices, edges and faces of some polyhedra ..... 22
1.6. Archimedean solids ..... 26
1.7. Origin of membrane processes ..... 29
1.8. Methods of preparation of membranes ..... 29
1.9. Probability density functions ..... 38
2.1. Symmetries of crystals ..... 60
3.1. Moments of 3-d Poisson-Voronoi tessellation, *cf* van de Weygaert (1994) ..... 79
3.2. Moments of 2-d section of 3-d Voronoi, *cf* van de Weygaert (1994) ..... 80
3.3. Statistics of 3-d VT, van de Weygaert (1994) ..... 80
3.4. Statistics of the planar section of the 3-d Voronoi tessellation, van de Weygaert (1994) ..... 81
3.5. Statistics for the line section of the 3-d Voronoi tessellation, van de Weygaert (1994) ..... 81
3.6. Statistics of a 3-d Voronoi network ..... 85
3.7. Neighbour statistics ..... 87
3.8. Three dimensions, statistics ..... 88
3.9. Results from 527 Voronoi cells ..... 92
3.10. Statistics of the number of vertices in high dimensions ..... 95
3.11. Two-dimensional Voronoi, 100 nuclei ..... 101
3.12. Three-dimensional Voronoi, 50 nuclei ..... 103
3.13. Four-dimensional Voronoi, 100 nuclei ..... 104
3.14. Four-dimensional Voronoi, 300 nuclei ..... 106
3.15. Five-dimensional Voronoi, 200 nuclei ..... 107
3.16. Intersection by a line, statistics ..... 109

3.17. Rayleigh distribution, line intersection statistics ..... 111
3.18. Mean of the cell surface area of compressed Voronoi ..... 114
3.19. Standard deviation of the cell surface area of compressed Voronoi ..... 114
3.20. Numerical values of the mean of the compressed cell volume ..... 115
3.21. Numerical values of the standard deviation of compressed cell volume ..... 116
3.22. Mean cell perimeter, numerical values ..... 116
3.23. Mean face perimeter, numerical values ..... 117
3.24. Standard deviation of the face perimeter, numerical values ..... 117
3.25. Cell and face perimeters, numerical values ..... 118
3.26. Numerical statistics of the area and perimeter of face ..... 121
3.27. The numerical values of face area ..... 122
3.28. Volume of $d$-hypercubes whose dimension is $\ell$ ..... 124
3.29. Voronoi volume in higher dimensions ..... 125
4.1. Percolation thresholds, Stauffer and Aharony (1998) ..... 129
4.2. Percolation probabilities, *cf* Sahimi (1994) ..... 130
4.3. Critical exponents and fractal dimensions, *cf* Sahimi (1994) ..... 130
4.4. Critical probabilities from Frisch *et al* (1961) ..... 138
4.5. Critical probabilities from Dean (1963) ..... 138
4.6. Critical probabilities and connective constants, Shante and Kirkpatrick (1971) ..... 140
4.7. Percolation statistics of the 2-homohedral tilings ..... 157
5.1. Particle size definitions ..... 178
6.1. Statistics of the modified Voronoi cells, Jafferali (1995) ..... 191
6.2. Comparative Voronoi statistics, Jafferali (1995) ..... 192
6.3. More comparative Voronoi statistics, Jafferali (1995) ..... 192

## List of theorems

1.1. Relationship between $\alpha$ and $r$ ..... 21
1.1[1]. Relationship between $A$, $V$ and $r_f$ ..... 22
1.2. Average number of edge per polygon ..... 43
3.1. Starting and ending nodes ..... 75
3.1[1]. Path along two dimensional Voronoi network ..... 76
3.1[2]. Path along three dimensional Voronoi network ..... 76
3.1[3]. Path along a two dimensional face of a three dimensional Voronoi network ..... 76
3.2. Vertices and edges, Euler ..... 85
3.3. Vertices in three dimensions, Tiyapan ..... 85
3.4. Edges in three dimensions, Tiyapan ..... 85
4.1. Setments always intersect ..... 141
4.2. Aggressive party ..... 141
4.3. Finding overlap ..... 142
6.1. The centre of gravity ..... 198

## List of algorithms

1.1. Perimeter per unit area of regular polygons ..... 20
1.2. Area per volume of regular polyhedra ..... 21
1.3. Thinning algorithm ..... 40
1.4. Triangulation edge length distribution ..... 42
1.5. Intersection between three planes ..... 45
1.6. Point inside a polygon ..... 46
1.7. Euclidean minimum spanning tree ..... 46
1.8. Rotation in three dimensions ..... 46
1.9. Ray and triangle intersection ..... 47
1.10. Convex hull in three dimensions ..... 48
1.11. Convex hull in two dimensions ..... 48
1.12. Quick hull algorithm ..... 49
1.13. Quaternary incremental method ..... 49
2.1. Generalised Johnson-Mehl model, Miles (1972) ..... 52
2.2. Drawing stereograms from crystal models ..... 55

2.3. Graham's scan ..... 63
2.4. Quickhull in two dimensions ..... 64
2.5. Convex hull of convex hulls ..... 64
2.6. Tilings in two dimensions ..... 72
3.1. Voronoi 2-d algorithm, Green and Sibson (1978) ..... 79
3.2. Valence relations ..... 100
3.3. Volume, area and perimeter algorithms ..... 120
4.1. Ant in a labyrinth ..... 129
4.2. Network percolation in 2-d ..... 132
4.3. Managing the Voronoi data in three dimensions ..... 134
4.4. Voronoi percolation in three dimensions ..... 135
4.5. Plane section of Voronoi in three dimensions ..... 136
4.6. Continuum percolation of regular polygons ..... 142
4.7. Threshold area ratio ..... 145
5.1. Fracture by attrition, Lim *et al* (1999) ..... 177
5.2. Arbitrarily shaped particles ..... 182
6.1. Voronoi construction in three dimensions, Tanemura *et al* (1983) ..... 185
6.2. Find the largest inscribed circle ..... 187
6.3. Interaction between particles and surface pores ..... 187
6.4. Interaction between each internal particle and the pores ..... 188
6.5. Particle cake formation by Jackson (1994) ..... 188
6.6. Flux decline across the membrane model ..... 189
6.7. The random walk of particles during removal by backflushing ..... 189
6.8. Non-woven fibre simulation, Wilkinson *et al*, 1986 ..... 190
6.9. Voronoi construction in 3-d ..... 190
6.10. Find faces of a polyhedron ..... 191
6.11. Nuclei points of the modified point process, Jafferali (1995) ..... 191
6.12. Asymmetric Voronoi tessellation, Jafferali (1995) ..... 191
6.13. Pore property evaluation, Jafferali (1995) ..... 192
6.14. Inscribed circle of faces ..... 193
6.15. Inscribed sphere of a polyhedron ..... 194
6.16. Verlet algorithm ..... 200
6.17. Filtration in Voronoi tessellation ..... 202
6.18. Voronoi data structure for the study of membrane filters ..... 202
6.19. Percolation by tiny particles due to attrition in membranes, preparing the structure ..... 210
6.20. Percolation by tiny particles due to attrition in membranes, percolation simulation ..... 211
8.1. Object location algorithm ..... 234
10.1. Beneath-Beyond algorithm ..... 350

## List of definitions

3.4. Volume of a $d$-simplex ..... 124
4.3. Threshold area ratio ..... 143

## List of assumptions

3.4. Volume of a unit cube ..... 123
3.4. Volume of $d$-simplices ..... 123

## Acknowledgements

## Notation

$0(n)$. a zero vector of $n$ dimensions.
$0(n,n)$. an $n \times n$ zero-matrix.
$1(n)$. a one vector of $n$ dimensions.
$1(n,n)$. an $n \times n$ one-matrix.
**a**. surface area of a cell.
$a^{ij}$. cofactor of $a_{ij}$, $a^{ij} = (-1)^{i+j} \det(a_i, b_i)$.
$\alpha$. (alpha) surface area per unit volume.
$\beta$. area of face.
$c$. concentration.
$d$. length.
$D$. diffusion coefficient.
$\delta$. thicness of the boundary layer.
$\epsilon$. porosity.
$\epsilon_0$. the permittivity of free space, $\epsilon_0 = 8.854 \times 10^{-12}$ $\mathrm{Fm}^{-1}$.
$\eta$. viscosity of liquid.
$e$. the charge of an electron, $e = -q$. See $q$.
$E(x)$. expected number of $x$
$f$. form factor. $f_c$, of a cell. $f_f$, of a face. $n$f, $n$-face.
**he**. he; he or she. $\mathrm{I}(n)$. an $n \times n$ identity matrix.
$I_r$. integers in the interval $r$ *eg*, $I_{[0,\infty)}$.
$J$. flux.
$\kappa$. Kurtosis.
$\mathcal{K}$. Kurtosis.
$\mu$. weight of particles in the cake per unit of filter surface; fluid viscosity; magnetic moment.
$\mu(\cdot)$. mean; also $\mu_g$ the geometrical mean, $\mu_h$ the harmonic mean.
$m^n(\cdot)$. the $n^{\text{th}}$-moment of.
$\mathrm{mad}(\cdot)$. mean absolute deviation.
$\mathrm{med}(\cdot)$. median.
$\mathcal{M}^n(\cdot)$. the $n^{\text{th}}$-moment of.
$\mathrm{N}$. original total number, for example $\mathrm{N}_c$ is the total number of cells, and $\mathrm{N}_v$ of vertices, created which may include those that are out of bound.
$n_x$. number of $x$. $n_c$, number of cells. $n_e$, number of edges. $n_v$, number of vertices.
$n_y^x$. number of $x$ of a $y$. $n_c^e$, number of edges of a cell. $n_c^f$, number of faces of a cell. $n_c^v$, number of vertices of a cell.
$n_{y,z}^x$. number of $x$ per $y$ of $z$. $n_{f,c}^e$, number of edges per face of a cell.
$\omega$. angular velocity.
$p$. probability; (with subscript) percolation probability. $p_c$, $p_b$, $p_v$, $p_e$, percolation probability of cells, bonds, vertices, and edges respectively. alternatively, $p_c^c$, $p_c^b$, $p_c^v$, $p_c^e$,.
$p(\cdot)$. probability density function, probability distribution.
$P$. pressure.
**Pc**. percolation. **Pc-process**, percolation process.
**PM**. *acronym* porous media; porous membranes.
$q$. the charge of a proton, $1.6 \times 10^{-19}$ C; feed rate.
$Q$. flow rate, $Q = \mathrm{d}V/\mathrm{d}t$.
w $r$:radius. $r_v$, $r_e$, $r_f$, radius to a vertex, edge, or face, respectively.
$\rho$. density; number density.
$r$. specific resistance.
§. (s) section or chapter.
$s$. perimeter, for example $s_c$ the perimeter of cell; cell perimeter.
$\sigma$. spin variable.
$\theta$. time.
$v$. velocity.
V. volume.
$W$. fluid velocity.
$x$. coordinates; coordination number.
$\bar{x}$. mean value of $x$, *cf* $E(x)$.
$z$. face perimeter.

## § 1. Introduction

This work began in 1995 under the co-supervision of now Emiritus Professor David J. Bell and Emiritus Professor Graham Arthur Davies. The is about the Voronoi tessellation and porous media. It is also about percolation. My objectives are to study the characteristics and nature of Voronoi tessellation, percolation, porous media which includes membranes, and look at their applications.

I started doing my master degree in Control and Information Technology in September 1994. I wrote a design exercise report on extremum control early in 1995 (§ E.4) and would have liked to, but could not do a dissertation on a similar topic. My dissertation, dated September of that year, was related to Voronoi tessellation and percolation. Then I went to Japan. There I studied singular perturbation, variable structure system and some other topics (*eg* §'s E.18, E.19 and E.20). I returned to Thailand in April 1999 and turned to teaching, translating and writing. In September 2000 I came to UMIST again to study translation (§ E.23) but later decided to resume the present research instead.

The procedure which I follow in doing the research is the following. During the course of the work I keep hand written notes which I call *work books*. All results exist in two places, either in the thesis or in the form of TEX document which I call *work notes*. Materials can move between the thesis and the work notes. Another group of work notes are those materials from the thesis which I consider as being key ingredients. These are written up in a form of brief reports or reporting materials. They are also filed for quick reference.

I call them *work notes* following the work notes of Michael Ventriswhich he kept in the course of deciphering the Mycenean Linear B script. I have kept work notes for one year and then discontinue the practice in order to brace myself towards finishing the present work. If ever I do resume the practice again for researches of my interest in the future it will probably be more compact than what I have done.

During the last year of my stay in Japan I kept my works together in what I call *technical reports*, the original of has been lost since I returned to England in 2000 because the original computer file kept on a computer in Japan has been lost. The name *technical report* is to follow that of the Technical Reports of the Control System Centre here at UMIST.

Topics are divided into Voronoi structure and geometry, percolation, and membrane science.

Voronoi tessellation has been used to model the array of the somata and cone cells in retina of mammals. (Ammermuller et al 1993; Ahnelt et al, 2000; Zhan et al, 2000) Curcio et al (1992) found anisotropy in cone spacing.

There have been theoretical works towards stochastic geometry. (Møller, 1989)

None of the papers and articles in § Ehas appeared in print elsewhere. Some have been translated into Thai by someone else and then published, but the English version included here is the original one. The paper in § E.17 should have been included in a conference proceeding but fortunately that is not the case, so it could be included here. Had it been published anywhere, I might have seen it suffices only to refer to the publication. As it is I see it fits to make sure it is here as complete as possible, if only because this is the only place where it is likely to be found in the future.

It may be true that nothing worthwhile is ever without trouble. This project has had its share of problems. I came to Manchester in September 2000 to do a Ph.D. in Translation, planning to wait, cook, translate, and demonstrate Thai martial arts to support myself. My supervisor was to be Peter Fawcett. I decided to do a Ph.D. with Graham Davies instead because of the following reasons: I was offered a financial support, the study would be two-year instead of three-year, and the topic of Technical Lexicon that Peter suggested did not interest me much. After having moved to the Chemical Engineering Department, I continued to help Mona Baker for a few months at her Corpus, and attended a few seminars organised by the translation group, until after the new year when I had to quit everything and concentrate on my Ph.D. work. After the Grosvenor Place where I used to stay direct-debitted my bank account over the amount of the rent by a few hundred pounds, I lost a few weeks trying to claim the money back. Then Nicholas Blackaby at the Information Service Department mistakenly killed about ten batch jobs that I had submitted to Cosmos, the loss of which set me back about one month in my work. Because of this, I never run a long job again, and has limitted my work only to those networks of smaller sizes. Then Graham Davies assigned me to

Leo Lue and retired. I was at sixes and sevens with Leo Lue's supervision. I still do not know what he wanted me to do; he never said the same thing twice. At the same time, there was a breakthrough in my ideas on languages and their clusters and connectivity. They are not acceptable in the strictly Chemical Engineering standpoint, so I had to fight to protect them like a father fighting to protect his children from being murdered.

Graham Davies started supervising me on this project in 1995. I completed a two-dimensional percolation program and used it to compute the critical probability of a Voronoi graph in time for my M.Sc. dissertation. I used the C program developed by Nicholas Jackson and maintained by Riaz Jafferali to generate the Voronoi network, and wrote my own program on Matlab to find the percolation probability. Before I went to Tokyo in September of the same year, we agreed that I would carry on doing some more work along the same line while I was in Japan. Graham suggested that I looked at the viscous fingering, a phenomenon which occurs when a low viscosity fluid penetrates a high viscosity one in the form of thin branches, which is of great importance in the replacement of oil in a reservoir by water.

The computers in Japan proved to be a nightmare for me for a long time. The technologies were there but when most things were in Japanese I could do next to nothing with them. I felt at a lost and still could not use them with confidence one year after I had arrived. From what I remember, there was one Mackintosh (!), two Linux's, and the remaining nine Sun's of various ages. But there was to be no Matlab until some two years later, by which time it was already too late for me to finish any investigation. Even the Sun machines were different because all the systems there have to be able to accommodate inputs in Japanese. System programmers in Japan have devellopped these systems so well that the two languages blend smoothly with each other; inadvertently press some certain key combination and you will never be able to get back to English again! My experience in Japan did not make me any wiser, and when I went to the University of Paris I at Sorborne–Panthéon in 2000 to do a DEA in Mathematics in Economics I was baffled there again by computers. Only that it was worse since all around me were Mackintosh machines and the french keyboard layout is different I did not wait long enough this time to get used to them.

In Thailand we have Thai Windows, but there are much less Thai facilities on Unix. Neither of them is anywhere near as well-devellopped as the systems in Japan, which may explain why the Japanese are further afield than us. Most people take these things for granted. It is something which is already there, one simply uses it. You only have to go to another country where a standard different from your own is used to see what an enormous amount of work must lie between the two systems. I still do not know how to tinker with a computer to make it digest Thai if it has not already been properly set up to do so. When I was in Japan I used to write articles for a journal of Thai students there. These I always wrote in English, and the publisher had them translated by somebody into Thai for the publication.

In this environment I telneted across half the globe from Tokyo to log on to the workstations at the Chemical Engineering Department in Manchester, but instead of writing another program to do viscous fingering I only managed to find the percolation probability of the road network in Bangkok using the program I from my M.Sc. I also carried further the work I had previously done with Martin Zarrop on object location using self-tuning control; and had an idea that if we could construct a network of people with an internal flow of money, we would be able to explain economic crises such as hyperinflation by using the percolation theory. On the topic of percolation and Voronoi tessellation I presented one paper at a conference in St. Louis (1996); on object location at two conferences, one in Atagawa (1996) and another one in Singapore (1997); and on economic modelling one at a conference in Tokyo (1997).

Here are the guidelines or requirements which Professor Davies has drawn up for my Ph.D. Firstly, to study the statistics of the three-dimensional Voronoi tessellation, namely the number of vertices, coordination probabilities of nodes, number of faces, volumes, area of each face, number of sides and edges of faces, and perimeter. Next, to 'formulate a simulation of Voronoi tessellation,' that is to say, polygons in two dimensions and polyhedra in three dimensions; then analyse the lattices for the number of cells, sides, and edges. In two dimensions find the fractional expected area per cell, and expected number of vertices and edges. In three dimensions find the expected number of vertices, edges, faces, the fractional expected volume per cell, area of faces and fractional area per cell. Then section the Voronoi tessellation in two dimensions using a straight line and repeat the same thing in three dimensions using a random plane. In two-dimensional section of three-dimensional lattices using orthogonal planes, find the number of cells, edges, and the area. And then compress the lattice in one plane, with compression ratios $0 \leq c \leq 0.8$, and find the volume

and surface area of cells for $c = 0$, 0.4, and 0.6; plot a section structure for $(\text{x–y})_{c_z=0}$, $0 < z < 1.0$, $c_y = 0$, 0.4, 0.6; $(\text{x–z})_{c_y=0}$, $0 < y < 1.0$, $c_z = 0$, 0.4, 0.6; compare the number of vertices, edges, and the area with those obtained from the case when $c_z = 0$, $c_y = 0$, and $z = 1$. Study transformations of Voronoi structures, for example the effect on the statistics when Voronoi graphs have a thickness. For each of the expected values mentioned above, find the first, second, and third moments. And finally, from the observation I have made that a three-dimensional tessellation have a bias towards an even number of vertices, find an explanation. To conclude the study of Voronoi tessellation, find affine transformations on two- and three-dimensional lattices and use the AVS program to print out sections of 3-d lattices †. Then for affine transformations, for example stretching, write algorithms to reposition the structure. For 2-d lattices, find the critical percolation probability (CPP), compare these with previous data and carry on to find the CPP on higher dimension lattices.

Graham mentioned as application to technical problems membranes, foams, plant cells, *etc*, and if possible to compare my results with models to describe other physical and natural phenomena. Relevant to the application in foams is perhaps the population balance equations. Care should be taken to ensure that any application mentioned is described in details how it can fit in with real problems. Of no less importance is to study methods of adjusting structures to fit the applications. These methods include modifications to Poisson point processes and, again, affine transformations of the cellular structure.

Filters used in aluminium smeltering must withstand a temperature higher than 700°Csince aluminium alloy melts at 650–680°C. These filters are used in order to separate the oxide froth which otherwise would form pits when the aluminium is casted. The process of manufacturing filters often introduces asymmetry elements to their structure, for instance there may be elongation in one dimension because of gravity, or in cases where material is drawn out to make the filter the structure may become distorted from the shear force of drawing. The texture of bread is anisotropic because the internal pressure from gas produced by the yeast press es the dough against its own surface tension. The elongated cells inside bamboos appear in various sizes, which explain why bamboos are at the same time strong and flexible. Certainly one sees the Voronoi tessellation wherever one looks. I feel that I have learnt so much from Graham.

The following summarises our original outline of the project. The objective is to study a *foam like* porous medium. Describe the statistics of structures in three-dimensional space, state the reason why we need the statistics and the roles they play in application. For the Voronoi tessellation, created from the Delaunay triangulation of Poisson points, give algorithms which generate the structure from one- up to $n$-space ‡, and give examples of these [structures found in nature]. Give the analyses of the static geometrical data, namely the length scale $l < L$ in one dimension; the area $a < A$, $n_1$, $n_2$, $L$ in two dimensions; and the volume $v < V$, $n_1$, $n_2$, $n_3$ in three dimensions, where $n_{d+1}$ is the number of the $d$-dimensional entities, $L = V^{\frac{1}{3}}$, $A = V^{\frac{2}{3}}$, and $V = \sum_i v_i$. Include the translation that I have made of the seminal papers by Voronoi and Dirichlet, a section on the transformations made on the structures, possible future developments of the project, and the applications.

Having given this thesis the first generic name, *Voronoi tessellation and porous media*, I kept in mind that it would probably have to be renamed with regard to the actual course of the devellopment. For this reason when asked by my supervisor Prof. Davies, I told him that he could help creating a new name, which he did *Division of space by Voronoi graphs, application to the models of porous membranes*. As this study, which has started off from the theory of percolation, has also led me to some interesting findings in fields other than Chemical Engineering, the name which I preferred used to be, *A twentieth-century definition of the ancient theme, Voronoi network and percolation in porous media*. In other words, *The TC definition of an ancient theme, VP in PM*. An alternative for the name used at present may be *On suspension blockages of filtering membranes as continuum percolation under van der Waals influences in centroidal Voronoi networks*

I have observed that a Voronoi cell in three dimensions always have vertices and edges in an integral multiple of two and three respectively. I found the reason for this and have given one proof for the first observation, and two for the second one (§ 3.7).

Any multilingual person looking at the development of clusters during the process of percolation will sooner or later see the similarity of this with the area of writing systems, that field which is half science as well as half language since it is essentially the same as a one-to-one mapping in mathematics.

---

† This has been done using Matlab instead.

‡ In one variation, $n \leq 5$.

One had better end his whole career if he wants to study percolation but would rather not look around for it in nature. Likewise if I were to leave out this connection which I have seen, it would be impossible for me to justify the inclusion in the appendix of the seminal works by Bravais, Dirichlet, Ising, and Voronoi which I have meticulously translated into English from the original texts in either German or French. If I knowingly refrain from mentioning this new idea in my publication it would amount to being unprofessional and untrue to my career in science, and I doubt if any university would let such a candidate graduate.

I have seen researchers who deliberately omit important things. For example one Ph.D. candidate I know who studies dancings of the Greek gypsies, who neither would mention the dancing for Dionysus nor believe in objective thinking in researchers (Theodosiou, 2002). Considering the role the dancing spirit of Dionysus plays in shaping the higher culture of the world, in particular the birth of the Tragedy, I could not help but think that such thesis as hers when finished could hardly be considered as complete or unbiased. The solemn dances for both the practice of weaponery and boxing of Ayudhàyạ, for instance, are very slow I find it difficult to think of them as dances. They are more like the breathing exercise practised by some pianists. But if I were to write *Dances in martial arts of the old Siam*, for instance, I would certainly discuss briefly in the introduction about the ritualistic dances of both the Zulu and the Dionysus, if only to point out where they are different and where the reader may find more information if he is interested to know more about them. It is true this may only be a matter of difference in philosophies of two researchers, but how can one be true to one's science unless one is true to oneself, and how can one be true to oneself unless one be faithful to one's philosophy.

When I found the roman writing for Thai that I think would work well, I had been thinking in Geometry most of the time for close to a year. It is difficult to prove the interrelationship between the two, but the working of the mind is not easy to fathom either. At that time I no more turned away from the Voronoi network to preoccupy my mind with language than I had an opportunity to say 'Good day!' to a person during the Christmas vacation of 2001 when the solution to this problem came to me. Aesthetics is the only remaining requirement other than the proof that the mapping is one to one, but this too seemed to me then to be immediately obvious that I considered the matter settled. It has been close to two hundred years since Kant made his call 'Sapere aude!', and if I dare not think for myself even at the present stage of my career then I probably deserve no Ph.D. of my own and the time elapsed since Kant was until now would prove to have come to nothing. Our course organiser Walter Korchinski had been wise to point out to me that I needed to find a more convincing argument to say that there is a relationship between language and percolation. I have since then seen more connections between the two than I had ever seen, and have become better convinced than I was before that it would be difficult for me to stop now.

Nor do I plan to neglect the Chemical Engineering application I have been assigned to do. I have looked very hard for any note received from Graham which guides the direction of the development of this project along the Chemical Engineering line. I have come to notice, and become convinced now, that he had never handed over to me any such requirement, not even about the interfacial forces which we have agreed verbally and about which he has so often talked. I have thought many times that I would remember to ask for these notes from him, but for one reason or another it always came to me too late everytime to ask for them, to the effect that most of the notes I have with me now are here only because he handed them to me himself since I seldom ever got around to asking for them. So many times I have received notes from him about the requirements regarding the Voronoi characteristics which seem a variation to the one I had previously received from him, such that I sometimes have up to four variations of practically the same note. This may also be because of my extremely slow pace at the resuming of the project in 2000 after a five-year gap, which resulted from the time needed for me to switch my mind back to thinking in Geometry again.

Leo Lue suggested the stress-strain analysis in the bonds of a Voronoi structure in three dimensions as well as the study of randomly placed rods. He also mentioned that Stuart Holmes has suggested the problem of percolation of randomly placed squares, which he thought would be useful in the growth process of synthetic zeolites. With him I could hardly work, for one thing I felt the need to *brag* to be accepted. If honesty be the best policy, then I should say that the only thing I really learn from Leo is that one has to boast to get by. My piano teacher used to tell me, 'Never belittle yourself! There are already many people out there who do that for you,' (Khitapanna, 2000) but still I find it distressing the need to protect yourself all the time against being belittled by your own supervisor.

The zeolite problem mentioned above I have found to be identical to the so-called *continuum*

*percolation* of squares. I have devellopped an algorithm to solve this not only for squares but also for the general case of n-gons. The program uses some of the ideas behind the basic movements in the Ayudhàya sword fighting. The algorithm has been intuitively arrived at from my own experience of several years holding a square, hind shield in one hand while in another brandishing a Thai sword for a troupe when our sword school demonstrated for that matter. It was only afterwards that I found the rigorous proofs for all its parts. However, I have not carried the work further to three dimensions because, for one thing, I personally do not believe the crystal growing to be random. Crystals grown within the same solution should have some kind of quantum coordination which allows them to align themselves with one another when they meet. The random appearance seems to be only their various habits.

I thank Sister Eileen and all the staffs at the St. Gabriel Hall of Residence who have made my stay there during the summer of 2002 a memorable experience. Special thank to Yann Frouin my flatmate at St. Gabriel for his encouraging words in times of turmoils, 'Believe in your ideas moreover they are beautifull.' [sic]

## § 1.1 Mathematics

A map, denoted by $f : \mathrm{X} \mapsto \mathrm{Y}$, is an assignment of an image $y$ in the target Y to a preimage $x$ in the source X, where $f$ is called a function and $f(x) = y$. A composition of $f : \mathrm{X} \mapsto \mathrm{Y}$ and $g : \mathrm{Y} \mapsto \mathrm{Z}$ is the map $h : \mathrm{X} \mapsto \mathrm{Z}$ such that $h(x) = g(f(x))$ for every $x$ in X. There are three types of function, each one having at least three names. Table 1.1 lists these three types together with the sources of their various names. The prefix *epi-* in Greek means *on* and *isos-* means *equal.*

| **Anglo-Saxon** | | **Greek** | | **Latin** | |
|---|---|---|---|---|---|
| an to an | one to one | monomorphos | monomorphism | injectus | injective |
| on to | onto | epimorphos | epimorphism | superjectus | surjective |
| an to an and on to | 1–1 and onto | isomorphos | isomorphism | bijectus | bijective |

**Table 1.1** *Three types of function and the source of their names.*

A concrete category is a set of function that is closed under composition and contains the identity map for every source and target. The maps in $C$ are the morphisms of a category $\mathcal{C}$, and both its sources and targets are objects. The topological category $\mathcal{T}$ is a category of topological spaces and continuous maps.

Furthermore, a functor is a map of categories which preserves composition and takes identities to identities. A covariant functor has $F(f \circ g) = F(f) \circ F(g)$, whereas a contravariant functor has $F(f \circ g) = F(g) \circ F(f)$. Homology is a covariant functor from $\mathcal{T}$ to the category of abelian groups and homomorphisms, while cohomology is a contravariant functor from $\mathcal{T}$ to the category of rings and ring homomorphisms.

A relation is an equivalent relation, denoted by $\sim$, if and only if three axioms, namely that of reflexivity, symmetry and transitivity are satisfied. A set of all $x$ in $A$ such that $x \sim a \in A$ is the equivalence clase of $a$. A partition of $A$ is formed by the set of equivalence classes of equivalence relations, and is a family of disjoint subsets of $A$ covering $A$. Then $A$ is the union of the equivalence classes, and each equivalence class has an empty intersection with any other. That is, every $a$ in $A$ is also in an equivalence class $E(a)$, and $E(a) \cap E(b) \neq \emptyset$ implies that $E(a) = E(b)$. Conversely, every partition of $A$ gives an equivalence relation on $A$, and $a$ is equivalent to $b$ if and only if $a$ and $b$ are in the same subset of the partition.

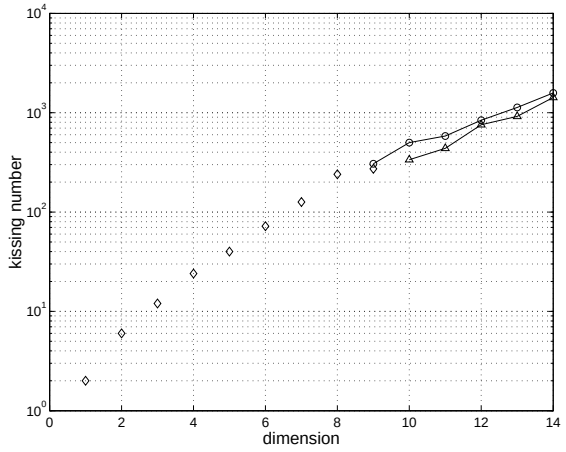
With the discovery of the equation $f_0 - f_1 + f_2$ by Euler in 1752 a new era of mathematic began, and that is the era of topology. The Euler's equation in its equivalent forms are written as $\sum_{i=0}^{d}(-1)^i f_i(P) = 1$ or $\sum_{i=-1}^{d}(-1)^i f_i() = 0$ (*cf* Grünbaum, 1967).

Partitions of numbers in the theory of numbers (*cf* Hardy and Wright, 1979) has a similar idea to what we shall do in § 6.11 for finding the Voronoi sections and their grids. Moreover, the random nature of these two problems, that is percolation and the number theory, as well as their omnipotence seem to point out that there could be some closer relationship between the two than mathematicians and physicists believe at present.

Also there is another similarity between these two fields, that is both are easy to state but very difficult to solve.

Sphere packing studies the density of packings of hard spheres. Here the kissing number problem studies the number of spheres that can be arranged such that all of them touch one central sphere of the same size. In other words it is the maximum number of neighbours that a sphere can have. It is also called the ligancy or Newton-, contact or coordination numbers. For $n$ dimensions the

lower bound for the kissing number is $\eta = \zeta(n)/2^{n-1}$, where $\zeta(n) = (1/\Gamma(n)) \int_0^\infty x^{n-1}/(e^x - 1)\mathrm{d}x$ is the Riemann zeta function.




In Figure 1.1 the a diamond symbol represents an exact value for a lattice, a triangle the lower bound for a lattice, while a circle the lower bound for a nonlattice. The lines linking these symbols are for the ease of look only.

**Figure 1.1** Kissing number versus dimension.

On the other hand, the covering problem is a problem that tries to find the least dense way to cover a space in $n$ dimensions with overlapping spheres of the same size while the quantising problem studies how to place points in such a way that minimise the average second moment of their Voronoi cell, which finds application in the conversion from analog to digital (Conway, 1988). In two dimensions the hexagonal lattice solves the packing, kissing, covering and quantising problems. But the packing problem of hard spheres is still unsolved. Everybody knows that the solution is approximately 0.7405 or exactly $\pi/3\sqrt{2}$ but no one has been able to prove it.

For any lattice the fundamental polytope, *i.e.* a polytope consisting of points $\sum_i^n \theta_i \nu_i$ where $0 \leq \theta_i < 1$, is its fundamental region. The proportion of space occupied by the spheres is the volume of one sphere divided by the volume of fundamental region, while the latter quantity is equal to $(\det \Lambda)^{1/2} = \det M$. Here $M = [\nu_{i,j}]$, $1 \leq \nu_{ij} \leq m$, is the generator matrix for lattice $\Lambda$ and each $\nu_i$ is a basis vector having $\nu_{ij}$ coordinates.

## § 1.2 Geometry

If the equation of a plane be $ax + by + cz + d = 0$, then $(a, b, c)^T$ is a vector normal to it and the parametric equations of this line is $x = at + p$, $y = bt + q$ and $z = ct + r$, $t \in R$, and $(p, q, r)$ is a point on the line.

The simplest geometrical figure is a circle while the simplest of all polygons is a triangle. The degree of freedom of triangles increases from the equilateral to the isosceles and the right triangles to the scalene triangles. While spending the summer of 1990 in a traineeship through AIESEC I was introduced to a geometrical puzzle which, as I came to learn later, is called the *flexatube*, but which I conjectured at the time that was from ancient China. This puzzle is made up of sixteen right isosceles triangles tiled into four squares, each comprising of four triangles, which are in turn joined together to form a loop. It can be easily made up using some hard papers, a pair of scissors and cello tape. There are in total twenty hinges, four of which are as long as the hypothenuse while the other sixteen have their length equal to the shorter side of the triangle. By turning these rigid triangles upon their hinges the inside surface of the strip can become the outside and vice versa. I found one solution a week later and back in Thailand a friend of mine found another solution. I think that these two are the only possible solutions but have not been able to proof it.

Having this interest in geometrical puzzles I was delighted at the time to find that the symbol for the men's toilets in Budapest is an equilateral triangle, while that for women's toilets is a circle. No other things beside these are written. Obviously these two geometrical forms suffice and are fully understood by the whole population.

The median of a triangle is the line joining its vertex with the mid point of the opposite side. All the medians of a triangle intersect one another at its centroid. The lines connecting all mid points of a triangle divide it into four equal triangles. Let the length of the median from the corner $A$ to its opposite side $a$ of a triangle be $m_a$, then $m_a = (2b^2 + 2c^2 - a^2)^{1/2}/2$, and similarly for $m_b$ and $m_c$. We have the relations $\mathrm{A}(\triangle m_a m_b m_c) = 3\mathrm{A}(\triangle abc)/4$, where $\mathrm{A}(\triangle\cdot)$ is the area of a triangle, and $m_a^2 + m_b^2 + m_c^2 = 3(a^2 + b^2 + c^2)/4$.

One end of a median is at a vertex, the other one at the mid point of a side. The lines which bound the first ones is the triangle itself, those which join the second ones create the medial triangle. A bisector of a medial triangle divides the perimeter of the original triangle into two equal parts. The incentre of the medial triangle, the Spieke point, is the c.g. of the wire-framed triangle $\triangle ABC$. The incentre, the geocentre and the Spieke point, all lie on a single straight line. The geocentre on the Euler line is one third away from the circumcentre to the orthocentre, that point where the altitudes of the triangle intersect.

In general dimension we talk about spheres. A sphere in $d$ dimensions has its volume, $V$, proportional to $r^d$ and its surface, $A$, to $r^{d-1}$, so that $V \propto A^{d/(d-1)}$.

The area of sphere in three dimensions is $A = 4\pi r^2$ and the volume $V = \frac{4}{3}\pi r^3$. When $V = 1$, $r = -(-3/\pi)^{1/3}/2^{2/3}$, $(3/\pi)^{1/3}/2^{2/3}$, or $(-1)^{2/3}(3/\pi)^{1/3}/2^{2/3}$ which are numerically $-0.310175 - 0.537239i$, $0.62035$, $-0.310175$, $-0.310175 + 0.537239i$ in that order. Therefore $\alpha = A|_{V=1} = 6^{2/3}\sqrt[3]{\pi} = 4.83598$

When $V = 8$, $r = (6/\pi)^{1/3}$ and therefore $A = 46^{2/3}\pi^{1/3} = 19.3439$. In order to find $\alpha$, the surface area per unit volume, one divide the area by $V^{2/3}$, in other words $\alpha = V^{1/3} \cdot A/V = A/V^{2/3}$.

The same is true for other polyhedra. For example in a tetrahedron where $x$ is the length of the side, the vertices can be $(0,0,0)$, $(x,0,0)$, $\left(\frac{x}{2}, \frac{\sqrt{3}}{2}x, 0\right)$, $\left(\frac{x}{2}, \frac{\sqrt{3}}{6}x, \frac{1}{2}\sqrt{\frac{93}{35}}x\right)$. When $V = 1$, one can obtain $x$ by solving the equation

$$1 = \frac{1}{6}\,\mathrm{abs}\left(\begin{vmatrix} 0 & 0 & 0 & 1 \\ x & 0 & 0 & 1 \\ \frac{x}{2} & \frac{\sqrt{3}}{2}x & 0 & 1 \\ \frac{x}{2} & \frac{\sqrt{3}}{6}x & \frac{1}{2}\sqrt{\frac{93}{35}}x & 1 \end{vmatrix}\right).$$

This gives $x = 2\left(\frac{35}{31}\right)^{\frac{1}{6}} = 2.0409$ as the only real positive answer. When $x$ is doubled, $V$ increases from 1 to 8, which means that one would be dividing $A$ by $V^{\frac{2}{3}}$ to obtain $\alpha$.

The perimeter of a triangle is $s = 3d_e$ and the area $A = (\sqrt{3}/4)l^2$. When $A = 1$, $d_e = \pm 2/3^{1/4} = \pm 1.51967$. Therefore $s = 4.55901$.

Vertices shared by two cells make up a common face between them. Two way have been tried for finding the edges. The first one was by looking at all neighbouring cells of every cell in turn three at a time. The edges are then made up of those vertices that are common among these three cells. Only those edges which have exactly two vertices are considered. They are called *good edges* as contrasted with edges on the boundary. This is a much longer way than the second one, which is to consider vertices common to any two faces of a cell. Similar to the first case, such vertices forms a good edge if and only if there are only two of them. The two methods above give exactly the same list of edges, so they confirm each other. It has been tested that all edges having more than two vertices are boundary ones, that is they have at least one vertex outside the boundary of the unit cube considered.

By drawing some of the cells as a solid using *fill* command it has been tested that the result from `convhull` covers the entire cell surface. This confirms the step where areas are calculated.

The hexagon or honeycomb is perhaps the pattern which is most frequently found in nature. Eventhough the world we live in is three-dimensional, cells normally divide and spread in two dimensions in the form of layers. Moreover, they are packed in these layers in patterns which most often resemble the honeycomb (*cf* Williams and Bjerknes, 1972).

An octagon is an eight-sided polygon. It is the shape of the cross section of every chimney in the mills built in Manchester during its industrial era of the nineteenth century, as well as that of the terrets in the Main Building of UMIST. Perhaps one of the reasons for its popularity is that it looks strong while having the style of a good taste. May be the reason why it looks strong is that it possesses eight axes of symmetry, on top of another symmetry around the origin.

There are nine regular polyhedra. Among these are five regular convex solids known to the ancient Greek called Platonic polyhedra. They are tetrahedron, cube, dodecahedron, octahedron, and icosahedron. They have regular congruent faces and regular polyhedral angle vertices. Their face angles and their dihedral angles at every vertex are equal. The other four regular polyhedra have only been discovered much later and are not convex. They are called the Kepler-Poinsot polyhedra and are nonconvex. The small stellated dodecahedron and the great stellated dodecahedron were found by Kepler (1571–1630). The great icosahedron and the great dodecahedron were found by Poinsot (1777–1859). The small stellated dodecahedron and the great dodecahedron do not satisfy

Euler's equation. The process of creating it by extending nonadjacent faces until they meet is called *stellating*. There are also polyhedra called quasi-regular.

The semi-regular polyhedra are called the Archimedean polyhedra. Here all faces are regular polygons but not all are of the same kind. Every vertex is congruent to all others. They comprise of an infinite group of prisms, an infinite group of antiprisms or prismoid, and another thirteen polyhedra. Each prism or prismoid is made up of two regular polygons on parallel planes where the vertices are aligned in the former case or shifted half way to the next neighbouring vertices in the latter case. Each vertex in prisms is joined to a corresponding vertex of the opposite polyhegon, while in prismoid it is joined to two corresponding vertices. All faces of an Archimedean solid are regular and all its polyhedral angle vertices congruent.

On the other hand the Archimedean duals have the property that all their faces are congruent to one another and all their polyhedral angles regular. These solids are important in crystallography. They are vertically regular and include an infinite group of dipyramids, an infinite group of trapezohedra, and additionally thirteen other polyhedra.

The surface area per unit volume $\alpha$ of a solid can be computed from the actual volume $V$ and the actual surface area $A$ as $\alpha = V^{1/3}A/V = AV^{-2/3}$. Another way of finding the perimeter per unit area of an $n$-gon follows the steps listed in Algorithm 1.1. Here $\theta$ is the angle made by the lines from the centre of gravity of an $n$-gon to its two consecutive vertices, $\alpha$ half the angle between two edges, $h$ the distance from the c.g. to each edge, that is the height of one of the $n$ identical triangles all of which have a vertex at the c.g., $a$ the area of each of such triangles, $A(d)$ the area of the $n$-gon in terms of the edge length, $s$ the perimeter and $d$ the edge length $d_e$. Algorithm 1.1, however, may be reduced to two steps, namely solving for $(nd^2/4)\tan[(n-2)\pi/(2n)]$ and then $s = nd$. Of course, if $n$ increases towards infinity then $s$ approaches $2\sqrt{\pi} \approx 3.5449$.

**Algorithm 1.1** *Perimeter per unit area of regular polygons.*

$\theta \leftarrow 2\pi/n$;
$\alpha \leftarrow (\pi - 2\pi/n)/2$;
$h \leftarrow (d/2)\tan\alpha$;
$a \leftarrow (1/2)dh$;
$A(d) \leftarrow na$;
**solve** $A(d) = 1$ **for** $d$
$s \leftarrow nd$; □

| **Polygon** | $n_e$ | $A$ | $s$ (numerical)] |
|---|---|---|---|
| Triangle | 3 | $(\sqrt{3}/4)d^2$ | 4.55901 |
| Square | 4 | $d^2$ | 4 |
| Pentagon | 5 | $5(1+\sqrt{5})d^2/\left[4(10-2\sqrt{5})^{1/2}\right]$ | 3.8119 |
| Hexagon | 6 | $3\sqrt{3}d^2/2$ | 3.7224 |
| Heptagon | 7 | $(7/4)d^2\tan(5\pi/14)$ | 3.6721 |
| Octagon | 8 | $2d^2\tan(3\pi/8)$ | 3.6407 |
| Nonagon | 9 | $(9/4)d^2\tan(7\pi/18)$ | 3.6198 |
| Decagon | 10 | $5\left[(5+\sqrt{5})/2\right]^{1/2}d^2/(-1+\sqrt{5})$ | 0.3605 |
| Undecagon | 11 | $(11/4)d^2\tan(9\pi/22)$ | 3.5944 |
| Dodecagon | 12 | $3(2+\sqrt{3})d^2$ | 3.5863 |

**Table 1.2** *Perimeter per unit area of n-gons.*

In a similar fashion the surface area and volume of regular solids can be found, but first we need to know more about these solids. Table 1.3 lists some of the important properties of regular solids. Here cyc$(\cdot)$ is the cyclical permutations and $\tau = (1+\sqrt{5})/2$, *i.e.* the golden ratio. Regular polyhedra can also be represented by Schlafi's symbol as $\{p, q\}$ where $p$ and $q$ are respectively the face- and vertex figures.

| **solid** | **aka** | $n_v$ | $n_e$ | $n_f$ | $x_v$ | $d_e$ | $r_v$ | $r_e$ | $r_f$ | **dual** |
|---|---|---|---|---|---|---|---|---|---|---|
| tetrahedron | 3\|2 3 | 4 | 6 | 4 | $(\pm1,\pm1,\pm1)$ even or odd -1's | $2\sqrt{2}$ | $\sqrt{3}$ | 1 | $1/\sqrt{3}$ | itself |
| cube | 3\|2 4 | 8 | 12 | 6 | $(\pm1,\pm1,\pm1)$ | 2 | $\sqrt{3}$ | $\sqrt{2}$ | 1 | octahedron |
| octahedron | 4\|2 3 | 6 | 12 | 8 | cyc$(\pm1,0,0)$ | $\sqrt{2}$ | 1 | $1/\sqrt{2}$ | $1/\sqrt{3}$ | cube |
| dodecah~ | 3\|2 5 | 20 | 30 | 12 | cyc$(0,\pm\tau,\pm1/\tau)$, $(\pm1,\pm1,\pm1)$ | $2/\tau$ | $\sqrt{3}$ | $\tau$ | $\tau[(\tau+2)/5]^{1/2}$ | icosahedron |
| icosahedron | 5\|2 3 | 12 | 30 | 20 | cyc$(\pm1,0,\pm\tau)$ | $2\tau$ | $\sqrt{2+\tau}$ | 1 | $(6+5\tau)^{1/2}/3$ | dodecah~ |

**Table 1.3** *Some important properties of regular solids.*

The ratio between the edge length and distance to face, the surface area, the volume and the ratio between surface area and volume of some solids are shown in Table 1.4.

| *solid* | $d_e/r_f$ | $A$ | $V$ | $\alpha$ |
|---|---|---|---|---|
| Tetrahedron $(3^3)$ | $2\sqrt{6}$ | $24\sqrt{3}r^2$ | $8\sqrt{3}r^3$ | 7.2056 |
| Cube, $(4^3)$ | 2 | $24r^2$ | $8d^3$ | 6 |
| Octahedron, $(3^4)$ | $\sqrt{6}$ | $12\sqrt{3}r^2$ | $4\sqrt{3}r^3$ | 5.7191 |
| Dodecahedron,$(5^3)$ | $\frac{4\sqrt{10}}{(3+\sqrt{5})(5+\sqrt{5})}$ | $300\frac{2}{5-\sqrt{5}}^{\frac{1}{2}}\frac{(1+\sqrt{5})r^2}{25+11\sqrt{5}}$ | $100\left(\frac{2}{5-\sqrt{5}}\right)^{\frac{1}{2}}\frac{(1+\sqrt{5})r^3}{25+11\sqrt{5}}$ | 5.3116 |
| Icosahedron, $(3^5)$ | $3(1+\sqrt{5})\left[6+\frac{5}{2}(1+\sqrt{5})\right]^{\frac{1}{2}}$ | $180\sqrt{3}\frac{(3+\sqrt{5})r^2}{17+5\sqrt{5}}$ | $60\sqrt{3}\frac{(3+\sqrt{5})r^3}{17+5\sqrt{5}}$ | 8.0484 |
| Sphere | $\infty$ | $4\pi r^2$ | $(4/3)\pi r^3$ | 4.8360 |

**Table 1.4** *Surface area per volume of regular solids.*

Algorithm 1.2 finds the values given in Table 1.4. Here $p$ is the face figure, $n = n_f$ the number of faces, $r = r_f$ the distance from the centre of the polyhedron to each face, $d = d_e$ the edge length, $s = A/V$ and $\tau = \left[1+\sqrt{5}\right]^{1/2}/2$ the golden mean. Furthermore, $a$ and $v$ are respectively the area of each face and the volume of the pyramid whose base is the face and the apex of which is the centre of the polyhedron.

**Algorithm 1.2** *Area per volume of regular polyhedra.*

$\theta \leftarrow 2\pi$;
$\alpha \leftarrow (\pi - 2\pi/p)/2$;
$h \leftarrow d\tan\alpha/2$;
$a \leftarrow pdh/2$;
$v \leftarrow ar/3$;
$A(r) \leftarrow na$;
$V(r) \leftarrow nv$;
**solve** $V = 1$ for $r$;
$s(r) \leftarrow A(r)/V(r)$; □

Since Algorithm 1.2 will be used to proof Theorem 1.1, we shall give it a formal proof.

**Proof.:** Each face of a regular polyhedron is a regular polygon by definition. The lines connecting the centre and c.g. of a face with all its vertices divide $2\pi$ radian into $p$ equal portions, where $p$ is the face figure. Each portion then represents the angle in radian made by two consecutive vertices of the face as seen from its centre. Let this angle be $\theta$, then it follows that $\theta = 2\pi/p$. From the symmetry around the c.g., all these angles together with the sides of the face form $p$ identical isosceles triangles whose two equal triangles are at the vertices of the face. Moreover, this angle, $\alpha$, is half the internal angle of a face vertex. Then, because we know that all the internal angles of a regular polygon sum up to $2\pi(n-1)/n$ radian, it follows that $\alpha = (\pi - 2\pi/p)/2$. The area of each face is therefore $a = p(d_e h/2)$, where $h = (d_e/2)\tan\alpha$ is the distance from the centre of the face to its edges. The volume of the pyramid which have the face as its base and the c.g. of the polyhedron as its apex is thus $v = ar_f/3$, and the polyhedral surface area and volume are respectively $A = n_f a$ and $n_f v$. Comparing the above with Algorithm 1.2 completes the proof. □

From the results in Table 1.4 we can see that the sphere has the ratio $\alpha$ less than every regular polyhedron; in fact one could conjecture that it has the smallest $s$ of all solids. Icosahedron, on the other hand, is a regular polyhedron which has maximised $\alpha$. So now we know, for instance, that a virus wants to maximise its surface area.

The ratio $\alpha$ here is not simply obtained by dividing the surface area by the volume of a solid, as Algorithm 1.4 also tells. We take the volume of a solid to be the unity first, then proceed to find its correspondent surface area. As an example to show that these two values are not the same, consider an icosahedron whose $d_e = 2\sqrt{2}$ and $r_f = 1/\sqrt{3}$. From the values of $d_e$ and $r_f$ the surface area and volume are respectively 362.765 and 37.8252, which result in $A/V$ being 9.5906 which is not the same as our $\alpha$.

The numerical ratios given in both Tables 1.4 and 1.4 are rounded approximates, to make them easier to read. The exact values can easily be obtained by following the steps of calculation in Algorithms 1.2 and 1.2.

After having worked with Algorithm 1.4 I noticed that $\alpha$ depends on $r_f$. A little investigation confirms this, and Theorem 1.1 arises as a result. A platonic solid is a regular solid and vice versa.

**Theorem 1.1.** *Let $\alpha$ be the surface area of a Platonic solid and $r_f$ the distance from its centre to each face. Then $\alpha r_f = 3$.:*

**Proof.:** There are five and only five such solids, therefore we find $\alpha$ of every one of them. Assuming that Algorithm 1.2 together with the resulting $A$'s and $V$'s shown in Table 1.4 are correct. Then simply divide each $A(r)$ by the corresponding $V(r)$ for the regular solids from the table to get $\alpha = A(r)/V(r) = 3/r$. □

Collorary 1.1 follows immediately from Theorem 1.1 and the definition of $\alpha$, but such interesting consequence as it is deserves being called a collorary.

**Corollary 1.1[1].** *In a regular polyhedron, $Ar_f = 3V$.:*

The duality among regular polyhedra is shown in Table 1.4. In particular, the tetrahedron is dual to itself, the octahedron and the cube are dual to each other and likewise the dodecahedron and the icosahedron. The icosahedron is a popular shape among viruses. As regarding duality, it is worth noting that all pyramid are self-dual.

Deltahedra are polyhedra which have all faces equilateral triangles. They have $2n$ faces, $3n$ edges and $n + 2$ vertices. There are eight deltahedra, namely regular tetrahedron (4 faces), triangular dipyramid (6), regular octahedron (8), pentagonal dipyramid (10), snub disphenoid (12), triaugmented triangular prism (14), gyroelongated square dipyramid (16) and icosahedron (20).

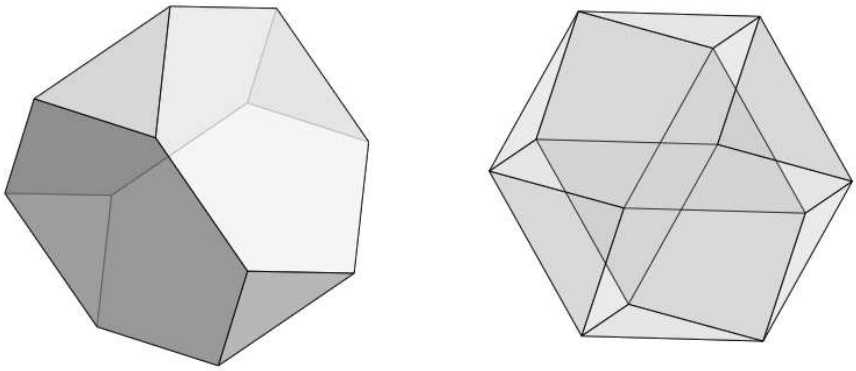
A cube is sometimes called a hexahedron because it has six faces. But there are other polyhedra which also have six faces, for instance the triangular dipyramid (5 vertices, 9 edges), the pentagonal pyramid and the tetragonal antiwedge (6, 10), the hemiobelisk and hemicube (7, 11) and the pentagonal wedge (8, 12).

| **polyhedron** | $n_v$ | $n_e$ | $n_f$ |
|---|---|---|---|
| antiprism | $2n$ | $4n$ | $2n+2$ |
| antiwedge | $2n-2$ | $4n-6$ | $2n-2$ |
| cupola | $3n$ | $5n$ | $2n+2$ |
| cupola pyramid | $3n+1$ | $7n$ | $4n+1$ |
| cupolarotunda (ortho-, gyro-) | $5n$ | $10n$ | $5n+2$ |
| deltohedron | $2n+2$ | $4n$ | $2n$ |
| dipyramid | $n+2$ | $3n$ | $2n$ |
| hemiprism | $2n-1$ | $3n-1$ | $n+2$ |
| ortho(,gyro-)bicupola | $4n$ | $8n$ | $4n+2$ |
| ortho(,gyro-)birotunda | $6n$ | $12n$ | $6n+2$ |
| prism | $2n$ | $3n$ | $n+2$ |
| pyramid | $n+1$ | $2n$ | $n+1$ |
| rotunda | $4n$ | $7n$ | $3n+2$ |
| rotundapyramid | $4n+1$ | $9n$ | $5n+1$ |
| wedge | $2n-2$ | $3n-3$ | $n+1$ |

**Table 1.5** *The number of vertices, edges and faces of some polyhedra*

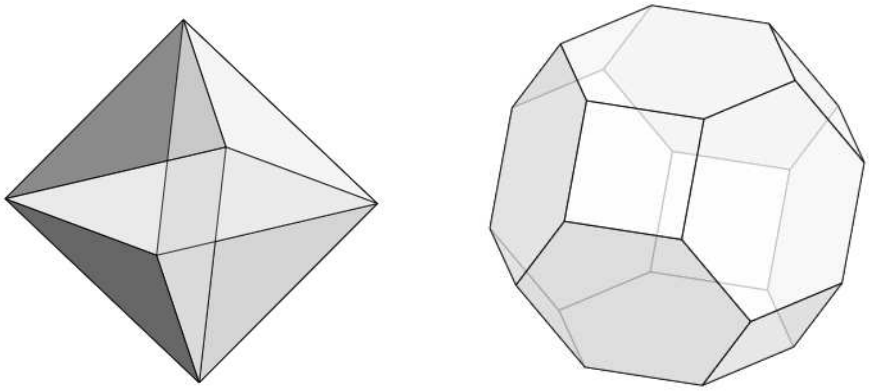
The nearest neighbour and minimum spanning tree have been applied to the problem of taxonomy in botany. Clayton (1972), working on the characters of plants to manually classify them (*eg* Clayton, 1970) with the use of only the binary dendrogram and trial and error, adopted a numerical method which finds the minimum spanning tree in a multi-dimensional character space. Since taxonomy can be considered as a kind of dictionary, it is possible to apply a similar approach to machine translation and the compilation of dictionaries.

The polyhedra from Figure 1.2 to 1.4 are semi-regular.



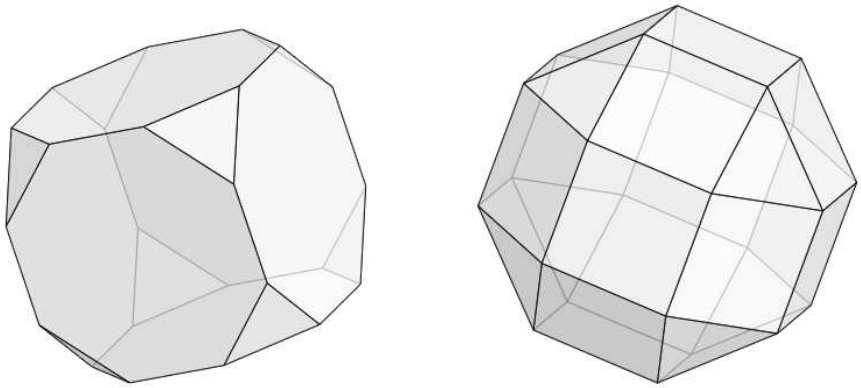
(a) (b)

**Figure 1.2** (a) *Truncated tetrahedron, triakistetrahedron,* 2 3|3. (b) *Octahemioctahedron, octahemioctacron,* 3/2 3|3



(a) (b)

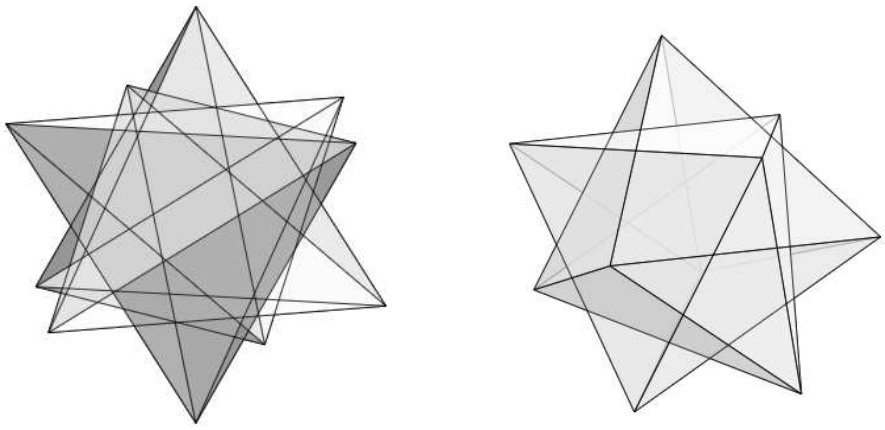
**Figure 1.3** (a) *Tetrahemihexahedron, tetrahemihexacron,* 3/2 3|2. (b) *Truncated octahedron, tetrakishexahedron,* 2 4|3



(a) (b)

**Figure 1.4** (a) *Truncated cube, triakisoctrahedron,* 2 3|4. (b) *Rhombicuboctahedron, deltoidal icositetrahedron,* 3 4|2

Polyhedra in Figure 1.5 are snub polyhedra.



(a) (b)

**Figure 1.5** (a) *Pentagrammic crossed antiprism, pentagrammic concave deltohedron,* |2 2 5/3. (b) *Pentagrammic antiprism, pentagrammic deltohedron,* |2 2 5/2

The surface of Fullerine is made up of pentagons six-sided figures. Its shape represents that of the geodesic domes developed by Buckminster Fuller, and hence the name *Fullerine*. The latter may either be hexagons or figures all the six sides in each one of which form two sets of three sides having an equal length. The simplest Fullerine, the carbon-60 molecule, has the same shape as that of a football and a handball. With some thought the reason for this is not difficult to see. With its thirty-two faces it closely resemble the sphere. Also the two different shapes of all its components are symmetrically distributed and therefore enable colouring with only two different colours, namely one for each of the two shapes. To see how this helps, suppose one made a football in the shape of a bloated dodecahedron. Then it would be impossible to colour it using more than one colour at the same time of giving it a symmetrical appearance when viewed from more than a few directions. With the fullerine shape and the colouring scheme mentioned, however, the football looks symmetrical when viewed from 54 different directions symmetrically distributed around it. These directions corresponds to those when one looks at it in the direction perpendicular to the centre of each of its faces and when in the direction through the middle of each of the 22 edges lying between two hexagonal faces.

Making polyhedron models is an educating experience. Contrary to the general believe that you need to make an accurate drawing for the required parts (Wenninger, 1971), this needs not be so. Examples of this are the origami models of polyhedra where complex polyhedron structures are made from interlocking pieces each of which is made by folding a piece of paper of a rectangular or square shape.

A set of elements with the sum and the product of any two elements defined is a commutative ring if under these two operations it satisfies the following postulates: closure, uniqueness, commutative, associative, and distributive laws, identity (zero and unity), and additive inverse. An integral domain is an ordered domain if its positive elements satisfy the laws of addition, multiplication, and trichotomy. A subset of an ordered domain is well-ordered if every nonempty subset of it contains a smallest member. $a|b$ means that $.b$ is divisible by $a$. The Euclidean algorithm or division algorithm states that $a = bq + r, 0 \leq r < b$. Two integers are relatively prime if their only common divisors are $\pm 1$. $a \equiv b (\mathrm{mod}\, m)$ if and only if $m|(a - b)$. The commutative ring $Z_2$ is the properties of multiplication and addition of even (0) and odd (1) numbers.

| + | **0** | **1** |
|---|---|---|
| **0** | 0 | 1 |
| **1** | 1 | 0 |

| · | **0** | **1** |
|---|---|---|
| **0** | 0 | 0 |
| **1** | 0 | 1 |

The following is $Z_5$.

| + | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **0** | 0 | 1 | 2 | 3 | 4 |
| **1** | 1 | 2 | 3 | 4 | 0 |
| **2** | 2 | 3 | 4 | 0 | 1 |
| **3** | 3 | 4 | 0 | 1 | 2 |
| **4** | 4 | 0 | 1 | 2 | 3 |

| · | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 1 | 2 | 3 | 4 |
| **2** | 0 | 2 | 4 | 1 | 3 |
| **3** | 0 | 3 | 1 | 4 | 2 |
| **4** | 0 | 4 | 3 | 2 | 1 |

There is a close link between geometry and algebra. Geometrical surfaces can be described

as algebraical equations. For example, for circles and polygons the equations are binary quadratic, while for spheres and polyhedra they are ternary quadratic. Even one-sided surfaces can be described algebraically. The equation of Klein bottle, when deformed into a sphere with two circles removed and replaced by two cross-caps, is a quartic equation

$$a^2(x^2+y^2)(b^2-x^2-y^2) = z^2(a^2x^2+b^2y^2),$$

while the Steiner surface is also a quartic one

$$y^2z^2+z^2x^2+x^2y^2+xyz = 0.$$

Two surfaces is homomorphic to each other if it is possible to continuously transform one into the other. All convex polyhedra are homomorphic to a sphere. The Steiner surface is homomorphic to the heptahedron, which is an Archimedean polyhedron with diametral plane.

In the plane, a second-degree equation gives either two straight lines, a circle, an ellipse, a parabola, or a hyperbola. In space, it can give two planes, cylinders and cones (circular, elleptic, parabolic, or hyperbolic), aphere, spheroid, ellipsoid, two hyperboloids, and (elliptic or hyperbolic) paraboloid.

*Partition*, *tessellation* and *division of space* are the same thing. In the context of set theory, a partition of set $X$ is a family of sets $A_1$, $A_2$, ..., $A_k$ which are subsets of $X$, such that $A_i \neq \emptyset$; $A_i \cap A_j = \emptyset$; $\bigcup_i A_i = X$, where $i$, $j = 1, 2, \ldots, k$ and $i \neq j$. (*cf* Berge, 1958) A further condition that makes any tessellation a Voronoi one is that, for all $i$ there exists a unique point $a_i$ within $A_i$ such that every point in $A_i$ is closer to $a_i$ than to any other $a_j$, $j \neq i$.

Voronoi tessellation in three dimensions can be constructed by imagining each region as a spherical cell growing outwards to meet neighbouring cells and continue growing to fill the gaps. The centre of each sphere is a unique nucleus point of the region such that it is closest to any point belonging to that region than any nuclei points. If the rate of growth is the same from every cell, the resulting partitions will be planes which can be described by ternary quadratic equations. However, if this rate differs from one cell to another, the partitions will be curved surfaces and the result is a non-Voronoi tessellation. It is possible to impose a constraint of minimum distance between neighbouring nuclei. Such cases can be looked at as spheres of an equal nonzero radius expanding away from nucleus centre points. If the radii differ from one sphere to another, or if some nonspherical solids are used instead of spheres, the tessellation obtained will be non-Voronoi.

Consider the case where all spheres are of equal size. If these spheres already touch their neighbours before the expanding starts, the case is that of packed spheres expanded to form a Voronoi tessellation. There are two types of close-packing: cubic (face-centred) and hexagonal. In both cases each sphere has twelve neighbours. Both cases have the same density, which is $\frac{\pi}{3\sqrt{2}}$. The Voronoi regions produced from the cubic case are rhombic dodecahedra and the faces are rhombuses. In the case of hexagonal close-packing, the corresponding regions are trapezo-rhombic dodecahedra and the faces are either rhombics or trapezia. Where the spheres meet with their three neighbours in the layer above and their three neighbours in the layer below, the faces are rhombics. Where they meet with the six neighbours on the same layer they are trapezia.

For geometrical calculation, an example of a definitive book is that written by Salmon (1912).

The gamma function,

$$\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}\mathrm{d}t, \quad \mathrm{Re}(z) > 0, \qquad (\mathbf{1})_\mathrm{i}$$

got its name from Legendre and is known as the Euler gamma function or simply the second Euler function. The formula $\Gamma(z+1) = z\Gamma(z) = z!$ recursively calculates the gamma function from, for instance, $\Gamma(1/5) \approx 4.5908$, $\Gamma(1/4) \approx 3.6256$, $\Gamma(1/3) \approx 2.6789$, $\Gamma(2/5) \approx 2.2182$, $\Gamma(1/2) = \sqrt{\pi} \approx 1.7725$, $\Gamma(3/5) \approx 1.4892$, $\Gamma(2/3) \approx 1.3541$, $\Gamma(3/4) \approx 1.2254$, and $\Gamma(4/5) \approx 1.1642$. The Stirling's formula was found by de Moivre which approximates the gamma function. The gamma function expansions is

$$\Gamma(x+1) = \lim_{k\to\infty} \frac{k^x 1\cdot 2\cdot 3\cdots k}{(x+1)(x+2)\cdots(x+k)}, \qquad (\mathbf{2})_\mathrm{i}$$

and the gamma function of negative numbers can be obtained from

$$\Gamma(-z) = \frac{-\pi}{z\Gamma(z)\sin\pi z}. \qquad (\mathbf{3})_\mathrm{i}$$

The incomplete gamma function is

$$\Gamma(z,x) = \int_0^x e^{-t}t^{z-1}\mathrm{d}t = \int_x^\infty e^{-t}t^{z-1}\mathrm{d}t, \qquad (4)_\mathrm{i}$$

and the normalised or regularised incomplete gamma function is $\Gamma(z,x)/\Gamma(z)$

Archimedean solids are dual to Catalan solids. The fullerene $C_{60}$ is the truncated icosahedron, an Archimedean solid. Table ℵ shows some statistics of the Archimedean solids.

| **Archimedean solid** | $n_v$ | $n_e$ | $n_f$ | **dual, Catalan solid** |
|---|---|---|---|---|
| truncated tetrahedron | 12 | 18 | 8 | triakis tetrahedron |
| cuboctahedron | 12 | 24 | 14 | rhombic dodecahedron |
| truncated cube | 24 | 36 | 14 | triakis octahedron |
| truncated octahedron | 24 | 36 | 14 | tetrakis cube |
| rhombicuboctahedron | 24 | 48 | 26 | deltoidal icositetrahedron |
| snub cube | 24 | 60 | 38 | pentagonal icositetrahedron |
| icosidodecahedron | 30 | 60 | 32 | rhombic triacontahedron |
| great rhombicuboctahedron | 48 | 72 | 26 | disdyakis dodecahedron |
| truncated icosahedron | 60 | 90 | 32 | pentakis dodecahedron |
| truncated dodecahedron | 60 | 90 | 32 | triakis icosahedron |
| rhombicosidodecahedron | 60 | 120 | 62 | deltoidal hexacontahedron |
| snub dodecahedron | 60 | 150 | 92 | pentagonal hexacontahedron |
| great rhombicosidodecahedron | 120 | 180 | 62 | disdyakis triacontahedron |

**Table 1.6** *Archimedean solids*

The analogue of polyhedra in four dimensions is sometimes called polychora, with the 4-d equivalent of the Euler-Descartes formula being $n_v - n_e + n_f - n_c = 0$ where $n_c$ is the number of its 3-d facets called cells. For dimensions higher than four the analogies are polytopes. An $n$-dimensional polytope is bound by hyperfaces of polytopes of $(n-1)$ dimension which join at hyperedges of $(n-2)$ dimensions.

The Euler characteristic $\chi$ is 1 for a point, invariant in a topological homeomorphism and additive for disjoint sets. For Euclidean space in $n$ dimensions, $\chi = (-1)^n$ as that of ordinary open polytopes. The $\chi$ of all ordinary closed polytopes is the $\chi$ of a closed $n$-dimensional ball, and $\chi = 1$ for any $n$ dimensions. If the hypercell itself, *i.e.* the interior of the polytope, is not counted then the right hand side of the formula becomes 2 when $n$ is odd and 0 otherwise, for example $n_v - n_e + n_f - n_c + n_t - n_p + n_h = 2$ in 7 dimensions, where $t$, $p$ and $h$ are respectively the tetrafaces (sometimes called hyperpoints), pentafaces (hyperedges) and hexafaces (hyperfaces).

As a revision of the sixth form mathematics, selections are *combination* if the order is irrelevant, and are *permutation* otherwise. The formula is for the former ${}^n\mathrm{C}_k = n!/[n!(n-k)!]$, and ${}^n\mathrm{P}_k = n!/(n-k)!$ for the latter.

Definitions which are useful when describing the time- and storage complexities of an algorithm are $O(f(n)) = \{g(n) : \exists c, n_0 \in R^+, g(n) \leq cf(n)\,\forall n \geq n_0\}$, $\Omega(f(n)) = \{g(n) : \exists c, n_0 \in R^+, cf(n) \leq g(n)\,\forall n \geq n_0$, $\theta(f(n)) = \{g(n) : \exists c_1, c_2, n_0 \in R^+, c_1 f(n) \leq g(n) \leq c_2 f(n)\,\forall n \geq n_0$ and $o(f(n)) = \{g(n) : \forall c \in R^+ \exists n_0 \in R^+, g(n) \leq cf(n)\,\forall n \geq n_0\}$, the most commonly used for the purpose being the $O(\cdot)$.

A geometry, in Klein's view, is a set $S$ and a subgroup $G$ of the group $B_{ij}(S)$ of all bijections from $S$ to itself. The elements of $S$ are points, and $G$ acts on $S$ by mapping points to points. Two subsets of $S$ are equivalent if there is an element of $G$ which takes one set into the other. In general, $S$ has an extra structure which $B_{ij}(S)$ preserves. A map $f : \mathrm{R}^m \mapsto \mathrm{R}^n$ is linear if it maps a linear combination of vectors to the same linear combination of the images. A matrix $(a_{ij})$, $1 \leq i \leq m$ and $1 \leq j \leq n$, transforms each basis element $b_i$ to a combination of the basis elements $b_j$. The map is a bijection if $f^{-1}$ exists or equivalently if the determinant of its matrix is non-zero. The general linear group, $GL(n,\mathrm{R})$, is the set of all invertible linear transformations from the vector space $\mathrm{R}^n$ to itself. The special linear group $SL(n,\mathrm{R})$, a subgroup of $GL(n,\mathrm{R})$, is the set of all invertible transformations with determinant 1. The orthogonal group, $O(n)$, the set of all orthogonal transformations $T$. The special orthogonal group $SO(n)$, a subgroup of $O(n)$, is the set of all orthogonal transformations whose matrix has determinant 1.

The group $l(\mathrm{R}^n)$ of all isometries of $\mathrm{R}^n$ consists of composites $T_a \circ L$, where $T_a : x \mapsto x + a$ is a translation and $L$ an orthogonal map. The set of all translations forms a normal subgroup of $l(\mathrm{R}^n)$ which is isomorphic to the group $\mathrm{R}^n$ under addition.

A summetry group of the subset $X$ of $S = \mathrm{R}^2$ is the subgroup of $l(\mathrm{R}^2)$ whose all elements map to themselves. Rotation by $1\pi/n$ generates a subgroup which is isomorphic to $C_n$, the cyclic group of order $n$. The dihedral group $D_n$, whose order is $2n$, is the group of symmetries of a regular $n$-gon. For $n \geq 3$, $D_n$ is a non-abelian group.

The affine group, $A(\mathrm{R}^n)$, is the group of all affine transformations or affinities of $\mathrm{R}^n$, that is to say, the transformation of the form $T_a \circ L$ where $T_a$ is a translation and $L \in GL(n, \mathrm{R})$. Affine transformations preserve no distance, angle, area or volume. But they preserve collinearity, parallelism and ratios. A similarity transformation or similitude, $T_a \circ \lambda L$ with $L \in O(n)$ and $\lambda > 0$, is an affine map which preserves angles.

Affine theorems are theorems which can be proved by only those concepts which are preserved by affine transformations. In other words, they are theorems which can be proved by vector methods without using norms, dot- or vector products. Examples of such theorems are the coincidence of the medians of a triangle, Ceva's theorem and Menelaus's theorem.

The homogeneous coordinates of a point $x$ on the affine line are $(\alpha, \beta)$, where $x = \alpha/bta$. Here a line $OA$ through the origin and a point $A$ on the line $y = 1$ is described by a vector $(\alpha, \beta)$ in it. The point at infinity has homogeneous coordinates $(1, 0)$. The projective line is denoted by $RP^1$.

The projective group, $PGL(n, F) = GL(n, F)/\{\lambda I | \lambda \in F - \{0\}\}$ where $F$ is any field, is the set of all projective transformations or projectivities. The standard reference points on $RP^1$ are $\infty$, 0 and 1. There is a unique projective transformation which takes any three distinct points to any other three distinct points. Let $a$, $b$, $c$ and $d$ be points in $RP^1$, and $\theta$ the map which takes $a$, $b$ and $c$ respectively to $\infty$, 0 and 1. Then the cross-ratio is $\theta(d) = (a, b; c, d) = (d-b)/(d-a) \cdot (c-a)/(c-b)$. This cross-ratio is preserved by projective transformations.

The general form of an algebraic surface is $f(x, y, z) = 0$l, where $f(x, y, z)$ is a polynomial in $x$, $y$ and $z$. A surface of order one is a plane, of order two a quadratic surface, for example ellipsoids and hyperboloids, and of order three a cubic surface.

## § 1.3 Physics

Percolation has been introduced and developed in the 1950s. (Hammersley et. al., 1954; Broadbent et. al., 1957) A typical physical problem which gives rise to problems in percolation is that of finding the probability that the centre of a sufficiently large porous stone gets wet when immersed in water. The internal structure of such a stone can be viewed as comprised of pores and solids. Water can seep through clusters of pores but is blocked when solids are encountered. Therefore the water can only reach the centre of the stone if there exists an open path, in other words a single cluster of pores, which leads it there. When the size of the rock is sufficiently large, this cluster is called an infinite cluster.

If on the other hand we consider the solids instead of the voids, we can reason that since the stone is rigid there have to exist at least one infinite cluster of solids. Then, because there must be rocks of various degrees of porosity which allow water to pass through, there must be a range of porosity which at one end is a nonpermeable stone where there exists an infinite cluster of solids but not an infinite cluster of pores, while at the other end it is the other way round, that is there exists an infinite cluster of pores but not one of solids. In other words, the latter case is the case where the rock has disintegrated into pieces. If one considers the pores and the solids as being two phases opposite to each other, then one can reason that the total volume of the two stays the same, whereas the ratio between them could vary.

The study of blockages in porous media can be translated into the study of percolation on the media when pores randomly turn into solids. In other words, it is the study of the formation of an infinite cluster of one phase within the infinite mass of the other. More often the latter infinite mass is taken for granted as always exists and remains the same. The domain of consideration is thus reduced to only the original pore spaces. Then, the study becomes that of inversion between two phases, namely the infinite cluster of free pores and the finite clusters of free pores. While the existence of an infinite cluster or finite clusters of blockages is of no consequences when one is only interested in the percolation point, it is considerably important when one wants to contemplate on the behaviours on either side of that critical point. It is important also when one studies changes in the rate of flow through the media.

One example is the study of traffic networks. Traffic congestion can be described as three degrees of flowability: free-flowing, congested, and stand-still. In a free-flowing traffic there are infinite clusters of roads and finite clusters of blocked roads. In a congested traffic there are both

infinite clusters of roads and infinite clusters of blocked roads. In a stand-still traffic there are no infinite clusters of roads, while there are infinite clusters of blocked roads.

Applications of the percolation theory includes the study of forest fires and epidemics, the study of defects in semiconductors, the study of the effective resistance of a disordered mixture of two materials, the study of the Curie point of spontaneous magnetisation in ferromagnets and the Ising model, and the study of the rigidity of networks.

Phenomena which contain phase transitions are normally related to percolation theory. Examples of physical phase transition includes the boiling point and the point of evaporation, the transition of materials into superconductivity state, the triggering of a chain reaction in nuclear fission, and the triggering of nuclear fusion of hydrogen burning in a new-born star. Apart from these, there are philosophical phase transitions which include the nature of understanding, the nature of scientific discoveries and technological progress, the making of mobs, revolutions and wars, enlightenment, and, of course, love. Nature is essentially made up of numerous local linear or gradual relationships bound together in a larger scale by abrupt transitions.

The word *percolation* is derived from the word *percolator*, a coffee making machine. The connotation is therefore that of fluid flowing through a porous medium. In dynamic and everyday applications the emphasis is usually on the fluid part, and one studies the point of transition when the flow becomes blocked. There is another side of the same system where one studies the structure of underlying network. In that case the emphasis is on the solid part, and one is interested to find the point where particles percolate to form a solid structure.

Thus the study of percolation falls broadly into two branches, the continuum percolation and the network percolation. The former finds applications in *crystallisation* where one would like to know the number density of crystals that will make, for example, a synthetic zeolite membrane; in *etching of metallic surface* where one would like to find the estimates of the shapes and sizes of the etch pits which overlap and form clusters according to etching time and concentration of the acid; or in *telecommunication networks* where one would like to determine the number of mobile phone cells which percolates a city. The latter finds applications in *filtration* where one would like to discard filters just before they become blocked or in *traffic networks* where one could compare the robustness to standstill between two traffic network configurations.

Phase transition occurs in various areas. Wherever it is seen, it has the same characteristic of an intensely exponential change in a short period within a long stretch of a much more gradual one. The rate of change at the point of transition is generally astronomical. The sudden switch from one phase to another found in the study of percolation is a phase transition. The assumption of the existence and occurence of infinite clusters as an explanation for these transitions can be justified by considering the promptness and the scale of theoretical transitions in a cosmological setting, namely the turning on of the general relativity at Planck time $10^{-43}$ seconds after the bigbang, the Grand Unified phase transition at $10^{-35}$ seconds and the start of nucleosynthesis at 1 second. (cf. Croswell, 1995) The temperature of the universe decreases exponentially and its trend against time has to be plotted on a log-log scale to be comprehensible for otherwise it would lie almost on top of the coordinate axes. Such a plot as shown in Figure ℵ, produced from the data from Vaas (2002), approximates a straight line with negative slope. The percolation at the big bang could possibly be a phase change between matter and antimatter.

There is an idea in cosmology which could easily be as old as the oldest known religion, that describes the universe as alternating in cycles of destruction and creation. One current theory holding that idea tries to explain the big bang as the collision between two universes in five dimensional space-time. My idea is that the universe undergoes phase changes periodically and percolates at the point the same one of which is differently called the big bang, the big crunch, or the big bounce depending how one looks at it. What we see now is only one side of the coin. We can not calculate pass the singularity at the big bang using our present theories because it is another phase complementary to us. That is why that creative singularity baffles us. But I have no doubt that a theory exists which can explain both or, if there are more than two, all phases as well as the transition point. And when we finally discover it we will understand the cosmic history and, hopefully, percolation.

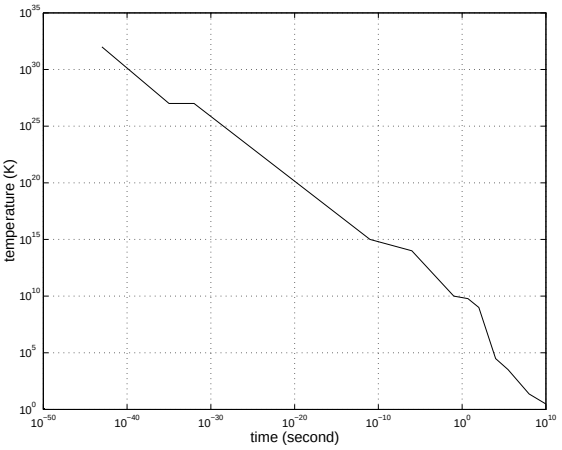



**Figure 1.7** Temperature of the universe according to the Big Bang theory..

In an asymmetric membrane the top dense layer determines the transport rate while the porous sublayer acts as a support. Because the permeation rate is inversely proportional to the thickness of the transport barrier, asymmetric membranes show a much higher permeation rate than homogeneous symmetric membranes of the same thickness.

Soap is solution of a sodium salt of a fatty acid, for example sodium stearate ($C_{17}H_{35}COO^{-}Na^{+}$.) The films of soap bubbles are made up of a monomolecular layer of amphipathic ions, which have a hydrophilic and a hydrophobic parts. The former forms a hydrophilic polar carboxyl head, which in the case of sodium stearate is the $COO^{-}$. The latter forms a tail of hydrocarbon chain, in this case $C_{17}H_{35}$.

| | Laboratorical scale | | Industrial scale | |
|---|---|---|---|---|
| *Process* | origin | application | origin | application |
| *microfiltration* | Germany, 1920 | bacteria filter | | |
| *ultrafiltration* | Germany, 1930 | laboratory use | America, 1960 | concentration of macromolecules |
| *hemodialysis* | Netherland, 1950 | artificial kidney | | |
| *electrodialysis* | | | America, 1955 | desalination |
| *hyperfiltration* | | | America, 1960 | sea water desalination |
| *gas separation* | | | America, 1979 | hydrogen recovery |
| *membrane distil -lation* | | | Germany, 1981 | concentration of aqueous solutions |
| *pervaporation* | | | Germany and Netherlands, 1982 | dehydration of organic solvents |

**Table 1.7** *Origin of membrane processes.*

There are three categories of synthetic membranes: porous, nonporous, and carrier. Micro- and ultrafiltration use porous membranes while nonporous membranes are used for dialysis, vapour permeation, gas separation and pervaporation. vAll membranes are required to be mechanically, thermally, and chemically stable. The means of separation, however, varies. For porous membranes it is the pore dimension, for nonporous membranes the difference in diffusivity or solubility, and for carrier membranes the properties of carrier molecules.

| **Methods** | **pore size ($\mu m$)** | **porosity** | **membrane types** | **examples** |
|---|---|---|---|---|
| **coating** | | | | |
| **phase inversion** | | | | |
| **sintering** | 0.1–10 | 10–20% | porous | |
| **stretching** | 0.1–3 | 90% max | porous | |
| **leaching** | 0.005 min | | porous | |
| **etching** | 0.02–10 | 10% max | porous | |

**Table 1.8** *Methods of preparation of membranes.*

In cosmology, the structure of the Abell Clusters shows a nonrandom distribution which suggests the existence of a second-order grouping, the clusters of clusters of galaxies (Abell, 1958). Apart from the Milky Way and the Andromeda galaxies (M31 or NGC224) there are at least 44 other galaxies in our Local Cluster. The Local Cluster is on the edge of the Local Supercluster which is flattened in shape and the approximate centre of which is at the Virgo Cluster. The Local Supercluster is separated by a nonspherical void of low galaxy density from the Pisces-Perseus supercluster. The latter lies nearly perpendicular to our line of sight, has the shape of a linear filament, and spans over 90° across the sky. The study of superclusters of galaxies often requires the use of superclusters of computers.

Abell (1958) plotted the distribution of the clusters in what he called, the 'Aitoff equal area projection'. By this he probably means the Hammer, aka Hammer-Aitoff, projection because that is what is normally used for astronomical maps in galactic coordinates as well as for maps of the whole world. Inspired by the Aitoff projection, it was created by H. H. Ernst von Hammer as a modification of the Lambert Azimuthal Equal Area projection. The Hammer projection is equal area while the Aitoff projection is not. Both are similar to each other in that both are modified azimuthal projections where the central meridian is a straight line half the length of the equator, the only point free from distortion is the centre point, and there is a moderate distortion throughout. My subsequent investigations lead me to conclude that it is a normal practice in Astronomy and Cosmology to call the Hammer projection as the Aitoff projection.

Powles and Quirke (1984) analyse the numerical trajectory of a molecule in liquid circles by using an empirical fractal parameter called the *Richardson coefficient*, $\alpha$. In their simulation they used the Lennard-Jones (12,6) intermolecular potential together with a reduced density and pressure. For the length of molecular trajectories in an argonlike liquid they found $\alpha = 0.65$, in comparison with a similar analysis done on a randomised Koch curve of order 6 which gives $\alpha = 0.25$ and the exact $\alpha = 0.2618\ldots$ for $K_\infty$, the Koch curve of an infinite order. The length of a fractal curve is $L(\epsilon) \propto \epsilon^{-\alpha}$, where $\epsilon$ is the step distance or scale. The graph between $\log(L(\epsilon)/\sigma)$ and $-\log(\epsilon/2)$ is a straight line with a positive slope represented by the equation $\log(L(\epsilon)/\sigma) = -\alpha \log(\epsilon/\sigma) + K$ where $K = \log k - (1+\alpha)\log 2$, $k$ being a constant. The Leonard-Jones (n,6) potential is the Van der Waals potential between two atoms which is described in the form $V(r) = C_n/r^n - C^6/r^6$ where $C_n$ and $C_6$ are constants, and $r$ is the distance between the centres of the two atoms. Similarly, the Leonard-Jones (12,6) potential is $V(r) = 4\epsilon\left((\sigma/r)^{12} - (\sigma/r)^6\right)$. It is sometimes written as $V(r) = \epsilon\left((R/r)^{12} - 2(R/r)^6\right)$, where $R = R_i + R_j$, $\epsilon = \sqrt{\epsilon_i \epsilon_j}$, $R_i$ and $\epsilon_i$ being the radius and respactively the interaction energy of each atom. In other words, in the last formula $\epsilon$ is the geometrical mean of the interaction energy of each atom while $R$ is the atomic cross-section which is a property of a pair of atoms.

## § 1.4 Cosmological structure

The Lorentz transformation was first written down by Voigt in 1887 as $x' = x - vt$, $y' = \frac{y}{g}$, $z' = \frac{z}{g}$, and $t' = t - \frac{vx}{c^2}$, then by Larmor in 1898, and finally by Lorentz in 1899. Poincaré stated in 1898 on the measure of time that there is no absolute equality of two time intervals. He named the Lorentz transformation after Lorentz and showed that together with the rotations they form a group. Working on the transformations, Einstein announced the special relativity in 1905 as a theory merited by its simplicity and beauty rather than being an explanation of experimental results. For the special relativity and the Lorentz transformations Minkowski founded a four dimensional non-Euclidean space to represent space-time which Einstein adopted used as a basis for his general relativity which appeared in 1915.

Whether the universe is homogeneous or isotropic depends on the scale in which one consider. If the scale is large enough then they can be considered both so but not otherwise. Similarly, an infinite Voronoi network originated from Poisson generators may be considered as being both isotropic and homogeneous because all the irregularities averages out.

Cosmology is related to the study of membranes and filters. Molecules of liquids in a membrane experience forces due to induced dipoles and fixed dipoles. The induced dipoles results in the Lennard–Jones potential whereas the fixed dipoles give rise to a permanent dipole moment between fluid molecules or between molecules of very fine particles.

Inter molecular interactions are approximated by considering the charged parts of the molecule as point charges. The force between charged parts of each molecule and those of its neighbours can be estimated by the Coulomb potential, $V = q_1 q_2/(4\pi\varepsilon_0 r)$, and the Coulomb force between each pair of molecules is $F = 1q_1 q - 2/(4\pi\varepsilon_0 r^2)$.

Analogous to the $1/r$, Coulomb potential in electrostatics is the $1/r$ potential in Swarzshield's expansion for the ten metric or gravitational potential of Einstein for the effect of an elementary concentration of mass in a space-time continuum that is asymptotically flat, that is

$$\mathrm{d}s^2 = \sum_{\alpha\beta} g_{\alpha\beta}\mathrm{d}x^\alpha \mathrm{d}x^\beta = \left(1 - 2Gm/(c^2 r)\right)\mathrm{d}r^2 + r^2[\mathrm{d}\theta^2 + \sin^2\theta \mathrm{d}\psi^2] - \left(1 - 2Gm/(c^2 r)\right)\mathrm{d}T^2. \quad (\mathbf{5})_{\mathrm{i}}$$

However, this is not as applicable as the Coulomb potential since the equations in the gravitation theory are nonlinear and therefore the superposition principle does not apply.

The Big Bang can be nothing but a change of phase of the universe. What the other phase may be we can not know, because there is a singularity which divides us from the Yonder Side. But one thing is indisputable, that is if we want to understand the universe, or in plainer words to make any sense out of it, we need to understand the singularity. And since we have to include the percolation theory in our final calculations, we might as well make it the beginning of our quest for a grand unified theory. Whether it is the percolation theory that we already know or another one not yet found is of no consequence. A percolation theory will still be a percolation theory no matter what form it may take, or indeed whatever name you may call it by. Our percolation theory is no percolation theory in a sense that it leaves out half of the picture, that is to say, the singularity that it still does not know how to explain. A true percolation theory is a theory which can includes singularities in its calculation while leaving out nothing that we know already. Give it some other name if you like, but *that* is the real percolation theory.

We now know many things in details, for example how stellar equations must account for mechanical, energy and thermal equilibria and that nuclear reactions imply conservation of charge, nuclear number and lepton number (*cf* Cooper *et al*, 1985). But at the Big Bang every one of these is supposed to break down. Because of this, our picture of the universe will always be incomplete until we can come to terms with that singularity which is our theoretical creator, the Big Bang.

## § 1.5 Filtration

Filtrationis the operation of separating a heterogeneous mixture of a fluid and particles of solids by means of a filtering medium which lets the fluid pass through but not the particles. The name filtration comes from the art of wine making. There are two processes involved, namely the flow of the fluid through the cake and the medium, and the filtration where particles are deposited on or in the medium. The objective is to understand how the rate of flow depends on the propeties of both the suspension and the medium, and on the operational conditions.

Both the cake and the medium are porous. As is the case with the cosmological structure (*cf* § ℵ), a filtering medium can be considered as being homogeneous on the large scale, whereas on the small scale it nearly never is (*cf* Heertjes, 1964). In other words, on the scale of the particles and the pores everything concerned is inhomogeneous to a high degree, that is to say, the slurry, the flow, and the cake. The micro-inhomogeneity in the cake can lead to a macro-inhomogeneity. Furthermore, the interaction among the particles, cake, medium and fluid makes the study of filtration ideal ground for numerical studies.

Heertjes (1964) describes the flow through a filter by the Fanning equation, $v = \mathrm{d}V/\mathrm{d}t = (1/\eta\gamma)\mathrm{d}(\Delta P)^{\delta}/\mathrm{d}R$, where $\eta$ is viscosity and $R$ the resistance. For viscous flow $\gamma = \delta = 1$, whereas for turbulent flow $\gamma = 0.11$ and $\delta = 0.55$. Both the cake ($c$) and the medium ($m$) have the say, so $R_c + R_m = R$, $\Delta P_c + \Delta P_m = P$, and $\Delta P_c/R_c = \Delta P_m/R_m$. The specific resistance of the cake is $r = \mathrm{d}R_c/\mathrm{d}w$, and therefore $R_c = \int_0^w r\mathrm{d}w = w(r)w$, where $w(\cdot)$ represents the mean value in contrast with $w$ which is the weight of particles in the cake per unit surface of filter. From slurry ($s$) the filtrate passes through the slurry-cake ($sc$-) and the cake-medium ($cm$) interfaces. A volume $V_{sc} = (1 + c/\rho_s)$ of slurry is needed to produce $V_{sc}$. Here $c$ is the concentration- and $\rho_s$ the density of particles. Assuming the amount of particles contined in the filter to be neglibible, then $w = cV_{sc}$. Then from $V_c = V_m$, $V_{sc} = V_m + V_R$, $V_R = w \cdot \bar{\epsilon}/[\rho_s(1 - \bar{\epsilon})]$ and $\epsilon_s = \rho_s/(\rho_s + c)$ we have

$$w = cV_m \,/\, \left[1 - \frac{(1 - \epsilon_s)}{\epsilon_s}\frac{\bar{\epsilon}}{(1 - \bar{\epsilon})}\right] \qquad (\mathbf{6})_{\mathrm{i}}$$

The filtration coefficient $\lambda$ in $-\mathrm{d}c/\mathrm{d}l = \lambda c$ is not a constant but change with time because of particles adsorbed by the bed. If one assumes that particles are bound to the wall by London-van der Waals force only, then $\lambda = K(\epsilon_0 - \sigma)$ where $\sigma$ is the specific deposit of solids in filter bed described as volume of solid per unit filter volume, $\epsilon_0$ the initial porosity of the bed and $K$ a function of London-van der Waals constant, $d$, $\epsilon$, $\eta$ and $W$. The equation of continuity is $W_0(\mathrm{d}c/\mathrm{d}l) = \mathrm{d}(\sigma)/\mathrm{d}\theta$. Cake is stabilised by the flow force and consolidates when it has reached a critical thickness and the velocity dropped below a critical value. The cake pressure is highest at the interface with the filter, so it is here that it starts to consolidate. Vibration is normally used to loosen it.

In the Diffusion Limited Aggregation model (*cf* Houi and Lenormand, 1986) particles $a_i$ are placed on the lattice while particles $b_i$ move towards them from a distance. These travelling particles stick to the first thing they meet, thus forming clusters, but they disappear whenever they stray too far away from the clusters. The density $\rho \sim nd^2/(hl) = na/(\tilde{x}l)$, $\tilde{x} = x/a$, approaches a fractal power law $\rho \sim \tilde{x}^{D-2}$ with the fractal dimension $D$ such that the deposit is homogeneous when $D = 2$ and heterogeneous when $D < 2$; $l$ is the length of the filter, $x$ the thickness of the deposit and $d$ the particle diameter. They consider two models, one to study the effects of random motion while the other that of ballistic trajectory. In the first model particles move in a square network, jumping from one site to one of its nearest neighbours with a probability $q$ towards the filter and $p$ in the other three directions. The diffusive- and the ballistic probabilities are respectively $4p$ and $1 - 4p$. The ratio between convective and diffusive displacement, a Péclet number, is defined as $P_e = (1 - 4p)/4p$. For the ballistic motion $P_e \to \infty$ while for the Brownian motion $P_e = 0$. In their second model particles move through space and $P_e = |u|/|r|$, where $u$ is a constant displacement vector in the flow direction while $r$ is in random direction. Particle A sticks to B when $\alpha < \beta$ or rolls on it if $\beta < \alpha < \gamma$. In this latter situation, A will stick to B if it is prevented from reaching the angle $\gamma$ comparative to A, but if at last $\alpha > \gamma$ A and B will separate.

The hydro dynamic forces act to transport particles through the medium. When the solid parts come close together, there is a viscous resistance which increases with the inverse of the separation between them. The van der Waals force acts at a close range, is always attractive and is theoretically infinite when particles touch a solid. Brownian motion affects particles smaller than $1\mu$m and results in a heterogeneous deposit, whereas the ballistic trajectory occured in sedimentation or filtration of big particles yield a deposit that is homogeneous.

Prefilters used to protect fibre bed coalescers from the damage caused by suspended solid

can become a bottle neck from being ladened with the solids itself. Chan (1990) was interested in such problem as the processing of hydrocarbon liquids on offshore platforms. In this process the four phase system, that of hydrocarbon gas, condensate, glycol and solids, is treated. The gas is separated and distillated to fractionate off butane and propane for uses in petrochemical manufacture. The other fluids contain hydrocarbon condensate and glycol which is added to prevent gas hydrates, a solid phase, from forming. But ethylene glycol has to be separated before it enters the purification system, for otherwise it would foul heat exchangers and the trays of distillation column. This can be conveniently done using a fibre bed coalescer. But solids suspended in one of the constituents, gas condensate glycol, will deposit and block these beds unless removed first by using a prefilter, which would then in turn become blocked and causes the bottle neck to the whole process.

Dead end filtration is simpler to simulate on the computer, but crossflow filtration is used more often in industry. Hydrophobic polymers in general have the advantage of good chemical and thermal stability, but hydrophilic polymers are becoming more and more attractive as membrane materials because they tend to have less adsorption. Adsorbed layer means more resistance to flow and a decline in flux. Moreover, these layers are difficult to remove by normal cleaning methods. Cellulose and its derivatives are among the best known hydrophilic polymers used as membrane materials. Examples of these are cellulose acetate, cellulose triacetate, cellulose tripropionate, cellulose nitrate, cellulose acetate-butyrate and ethyl cellulose. They are used in micro-, ultra- and hyperfiltration as well as in dyalysis and gas separation. Cellulose is hydrophilic but not soluble in water. It has a regular chain structure and is quite crystalline. Cellulose nitrate and cellulose acetate are used in micro- and ultrafiltration. Cellulose esters have excellent membrane properties except with regard to their sensitivity to biological-, chemical- and thermal degradations. They are made by air-casting or dry phase inversion. In phase inversion, a polymer is transformed from a liquid- to a solid state. Solidification is often started by liquid demixing, the transition of one liquid into two liquids. Phase inversion techniques include evaporation and precipitation the widely used for membrane of which is immersion precipitation (*cf* Schumacher, 1996). The top layer of an asymmetric membrane is dense and therefore responsible for most of the filtration. The sublayer is porous and provides the support to the top layer. Macrovoids are often found in the sublayer which lead to weak spots in the membrane and must be avoided, especially in high pressure applications which use dead end filtration. Tetrahydrofunate and acetone give membranes with a dense top layer as the result of delayed demixing. Dimethylsulfoxid and Dimethylformiade give membranes with a more porous structure from instantaneous demixing. Analogous to this is how pumice is very porous from the instantaneous solidification when it forms.

During the separation the flux through the membranes declines or decreases with time from adsorption, concentration polarisation, fouling, gel layer formation and pore pluggings. This is especially severe in micro- and ultrafiltration, with the decline in the flux often exceeding ninety per cent.

Darcy's law gives the volumetric flux of a pure liquid through a membrane, $J = \Delta P/(\eta R)$ where $R$ is the overall resistance of the membrane which includes the resistances from adsorbed particles, cake, concentration polarisation, gel, pore blocking and membrane. Adsorption decreases the pore radius according to the Hagen-Poiseuille equation, $\Delta r/r = 1 - (J/J_m)^{1/4}$. At steady state the convective transport balances the permeate flow past the membrane and the diffusive back flow which results from the accumulation of solute at the membrane surface, $Jc + D\partial c/\partial x = Jc_p$ where $D$ is the diffusion coefficient. Then from the boundary conditions $c = c_m$ at $x = 0$ and $c = c_b$ at $x = 1$, $(c_m - c_p)/(c_b - c_p) = \exp(J\delta/D)$. In other words, $c_m/c_b = \exp(J/k)/[R_n + (1 - R_n)\exp(J/k)]$ where the mass transfer coefficient $k = D/\delta$ and the intrinsic retention $R_n = 1 - c_p/c_m$. When the solute is completely retained by the membrane, $R_n = 1$ and $c_p = 0$, and therefore $c_m/c_b = \exp(J/k)$.

Across the filtering medium there is a driving force, in other words the pressure drop. There are four driving forces, centrifugal, gravity, pressure and vacuum. Filters in use in practice are either surface- or depth filters. In the former the solids are deposited on the surface in the form of a cake, thus the name *cake filtration*, while in the latter they are deposited inside the medium, thus *deep bed filtration.*

In filtration, Darcy's law is often written $Q = A\Delta p/(\mu R)$, where $R = L/K$ is the medium resistance, $L$ the thickness- and $K$ the permeability of the bed. If there is a cake, $R = R + R_c$ where $R_c$, the cake resistance, is $R_c = \alpha w$, $\alpha$ being the specific cake resistance in $\mathrm{mkg}^{-1}$ and $w$ is the mass of cake deposited per unit area. Cakes are normally compressible, so $\alpha$ changes with $\Delta p_c$ and is approximated as $\alpha_a$, where $1/\alpha_a = (1/\Delta p_c)\int_0^{\Delta p_c} \mathrm{d}(\Delta p_c)/\alpha$. There is an experimental empirical relation $\alpha = \alpha_0/(\Delta p_c)^n$, where $n$ is the compressibility index, and $\alpha_a = (1 - n)\alpha_0(\Delta p_c)^n$.

The mass of cake deposited is $wA = cV$, where $c$ is the concentration of solids in the suspension. For incompressible cake, $Q = \Delta pA/[\alpha\mu c(V/A) + \mu R]$ or equivalently $\mathrm{d}t/\mathrm{d}V = \alpha\mu cV/(A^2\Delta p) + \mu R/(A\Delta p)$. If $\Delta p$ is constant, $t = a_1V^2/(2A^2\Delta p) + b_1V/(A\Delta p)$, where $a_1 = \alpha\mu c$ and $b_1 = \mu R$. The experimental determination of $\alpha$ and $R$ works with this equation in the form $t/V = aV + b$, where $a = a_1/(2A^2\Delta p)$ and $b = b_1/(A\Delta p)$; or rather in a more detailed form $(t - t_s)/(V - Vs) = \alpha\mu c(V + V_s)/(2A^2\Delta p) + \mu R/(A\Delta p)$, where $t_s$ is the starting time at the beginning of the truly constant pressure period.

In constant rate filtration, $Q$ is kept constant, $Q = \Delta p(t)A/[\alpha\mu cV(t)/A + \mu R]$. In other words, $\Delta p = \alpha\mu cQ^2t/A^2 + \mu RQ/A$, or $\Delta p = a_1v^2t + b_1v$, where $v = Q/A$ is the approach velocity of the filtrate.

In many cases, operation demands constant rate followed by constant pressure, in which case $\Delta p = a_1v^2t + b_1v$ for $t < t_s$ and $\Delta p = \Delta p_s$ a constant for $t \geq t_s$. This amounts to $V = Q_1t$ for $V \leq V_s$ and $(t - t_s)/(V - V_s) = a(V + V_s) + b$ for $V > V_s$.

If a centrifugal pump is used, it is the case of variable pressure and variable rate. Here the equation is $V = A(\Delta pA/Q - \mu R)/(\alpha\mu c)$, and the filtration time necessary is $t = \int_0^V \mathrm{d}V/Q$.

For compressible cakes, $\Delta p = \Delta p_c + \Delta p_m$, where $\Delta p_m = \mu RQ/A$ and $\Delta p_c = \alpha_a\mu cVQ/A^2$. And $\Delta p_c = (1-n)\alpha_0\Delta p_c^n\mu cVQ/A^2$, that is to say, $\mu cVQ/A^2 = (\Delta p_c)^{1-n}/[(1-n)\alpha_0]$. Then, for the constant rate filtration, $(\Delta p_c)^{1-n} = \alpha_0(1-n)\mu cQ^2t/A^2$. And for the variable pressure and variable rate operation, $V = \left[A^2/((1-n)\alpha_0\mu c)\right] \cdot \left[(\Delta p - \Delta p_m)^{1-n}/Q\right]$, where $\Delta p_c$, $\Delta p_m$, $V$, $Q$ and $t$ are all variable.

The relationship between the specific cake resistance, porosity and specific surface is the Kozeny-Carman equation, $\alpha = K_0S_0^2(1-\varepsilon)/(\rho_s\varepsilon^3)$, where $K_0$ is the Kozeny constant which is approximately 5 for the lower porosity ranges, $S_0$ is the specific surface of the particles making up the bed, that is the ratio between the surface area and the volume of solids, $\rho_s$ the solid density and $\varepsilon$ the porosity, that is the ratio between the volume of voids and the volume of cake.

Fluid within porous media is essentially stagnant and the flow is laminar. Perhaps second only to the Hubble's constant in Cosmology in the matter of elusiveness is the $k$-factor which accounts for the tortuosity in porous filters, which, according to Kozeny and Carman, is $k = k_0(L_e/L)^2$, where $k_0$ is the shape factor, $L_e/L$ the tortuosity factor, $L_e$ being the interstitial length followed by the streamline and $L$ is the thickness of the bed. There are many other formulae (*cf* Piekaar and Clarenburg, 1967), for instance $k = k_0/\rho$ suggested by Sullivan where $\rho = \overline{(\sin^2\phi)}$ is the orientation factor.

Transport mechanism in filters is due to difffusion, gravity and hydrodynamic force. The efficiency is minimum at about the particle size of 1 $\mu$m. Filter normally runs about 24 hours between washes, with a rate between 5 and 15 $\mathrm{m \cdot h^{-1}}$. Wash rates are approximately 0.5 $\mathrm{m \cdot min^{-1}}$. (*cf* Ives, 1977).

## § 1.6 Statistics

Poisson distribution, defined by $p(x,\lambda) = \left[\lambda^x e^{-\lambda}/x!\right] \mathrm{I}_{[0,\infty)}$, is the binomial distribution, $p(x,n,p) = {}^n\mathrm{C}_x\theta^x(1-\theta)^{n-x}\mathrm{I}_{[0,n]}$, when $n$ goes to infinity, $\theta$ goes to zero, while $n\theta = \lambda$. Here $\theta$ is the probability of success of each trial. It is used when counting the number of occurences of a random event. Analogously Poisson point process, which has $p(x = n(v)) = \left[\lambda|v|e^{-\lambda|v|x}/x!\right] \mathrm{I}_{[0,\infty)}$, is the binomial point process, $p(x = n(v)) = {}^n\mathrm{C}_x\theta^x(1-\theta)^{n-x}\mathrm{I}_{[0,n]}$, when the volume $V$ goes to infinity, while $n/|V| = \lambda$. Here $\theta = |v|/|V|$ is the probability of points within $V$ being placed in $v \subset V \subset R^d$, and $\lambda$ the density or intensity of points. Therefore the density of point of a Poisson point process is constant by definition. A point process is a procedure which generates points on a domain within a space of $d$ dimensions.

The Poisson point process thus derived has the properties that $0 < p_{n(v)=0} < 1$ for $0 < |v| < \infty$, $\lim_{|v|\to 0} p(n(v) \geq 1) = 0$, $n(v_i)$ mutually independent and $n(\bigcup_n v_i) = \sum_n n(v_i)$ when $A_i$ are disjoint, and $\lim_{|v|\to 0}\left[p(n(v) \geq 1)/p(n(v) = 1)\right] = 1$.

The weighted mean of a group of data is $\mathbf{x} = \sum_i f_i x_i/n$ and the weighted variance is $\sigma^2 = \sum_i f_i(x_i - \mathbf{x})^2/n$, where $f_i$ is the occurrence frequency of $x_i$ and $\sum_i f_i = n$. Likewise the $r^{\text{th}}$-moment around the average is $m_r = \sum_i f_i(x - \mathbf{x})^r/n$, while the $r^{\text{th}}$-moment around the origin is $m'_r = \sum_i f_i x^r/n$ (*cf* Spiegel, 1975). Some relations among these various moments are $m_1 = 0$, $m_2 = m'_2 - {m'_1}^2$, $m_3 = m'_3 - 3m'_1m'_2 + 2{m'_1}^2$, and $m_4 = m'_4 - 4m'_1m'_3 + 6{m'_1}^2m'_2 - 3{m'_1}^4$.

The variance when normalised by $n-1$ gives the best unbiased estimated variance if the sample has a normal distribution. On the other hand the variance which is normalised by $n$ is identical with the second moment of the sample about its mean.

The log-normal distribution is closely related to the normal distribution. If $\ln x$ has a normal distribution with $\mu$ and $\sigma^2$, then $x$ has a log-normal distribution with $\mu$ and $\sigma^2$. In other words, a log-normal distribution curve will appear as a normal curve when plotted with a log scale in the $x$ axis. Its probability distribution function is $f(x|\mu,\sigma) = \left[1/(x\sigma\sqrt{2\pi})\right] \exp\left[-(\ln x - \mu)^2/(2\sigma^2)\right]$. It is positive definite, and therefore attractive in some areas of application where this is required, for example the amount of rain fall or particle size distribution. Both the log-normal and the normal distributions require only two parameters to describe, that is its mean and variance.

The most common drop size distribution in agitated heterogeneous liquid-liquid systems are the normal and the log-normal distributions (Giles *et al*, 1971). A straight line is obtained when the diameter of drops, $d$,with normal size distribution is plotted against cumulative percentage frequency, or in the case where they have log-normal size distribution, when $\log d$ is plotted against cumulative percentage frequency.

Monte Carlo methods (*cf* Hammersley and Morton, 1954) use random numbers. They have found applications in a wide variety of fields, ranging from numerical analsis to recreation. Random numbers sometime come from nature, for instance the generator which creates random numbers from a resistance noise suggested by A. M. Turing, in which case they are truely random in nature. In 1927 L. H. C. Tippett compiled a table of random numbers, and in 1955 RAND Corporation (*cf* Knuth, 1998). For the premium bonds lottery ERNIE is used in the UK since $2^{nd}$ June 1957 to generate a 9-digit sequence of random numbers by exploiting the random frequency instability in a free-running oscillator. ERNIE was upgraded in 1973 and the present version, introduced in 1988, is Mark 3.

Normal distribution is a family of curves which have two parameters, namely the mean $\mu$ and the standard deviation $\sigma$. The standard normal distribution, $\Phi(x)$, has $\mu = 0$ and $\sigma = 1$. It is related to the error function by the relation $\mathrm{erf}(x) = 2\Phi(x\sqrt{2}) - 1$. The central limit theoram states that, as the sample size increases to infinity, the sum of independent samples from some distribution of finite mean and variance converges to the normal distribution.

## § 1.7 Poisson process

The Poisson process is the probability model with one parameter, which represents all processes in which points occur randomly in time. The gamma experiment is to run the process to find the time $t_k$ of the $k^{\text{th}}$ arrival, whereas the Poisson experiment is to run it to find the number of arrivals $n_t$ in the interval $(0, t]$, $t \geq 0$. These two experiments give rise to two sets of random variables dual to one another, and $n_t \geq k$ if and only if $t_k \leq t$. The regeneration property says that the process after any time $t$ is independent of the process before $t$ and is probabilistically the same as the original process. The interarrival time is $x_1 = t_1$, $x_k = t_k - t_{k-1}$ for $k = 2, 3, \cdots$.

In a Poisson process the number of changes in each of the non-overlapping intervals is independent from that in the others. Let $\nu$ be the number of one change, $h = 1/n$ a sufficiently small interval and $n$ the number of trials. Then the probability of exactly one change in $h$ is $p = \nu h = \nu/n$. The probility of two or more changes in $h$ is zero. The number of $k$ changes occuring in $n$ trials is the continuous limit of the discrete binomial distribution

$$p(k) = \frac{n!}{k!(n-k)!} \left(\frac{\nu}{n}\right)^k \left(1 - \frac{\nu}{n}\right)^{n-k} \qquad (7)_{\text{i}}$$

## § 1.8 Phase transition

In the Ising model each spin has two possible states, that is up and down, and the hamiltonian is $H = J_0 \sum_{<i,j>} \sigma_i \sigma_j$ where the summation is over the nearest neighbours. Since it has been exactly solved, the Ising model provides a good model for the understanding of phase transition. This model can represent the transition from ferro- to paramagnetic at the critical temperature where the correlation lenght becomes infinite. Characteristic to the Ising model is the peak in the specific heat at the critical temperature.

The two-dimensional $xy$ model is a model of spins confined to a plane, the hamiltonian of which is $H = J_0 \sum_{<i,j>} \cos(\theta_i - \theta_j)$. This model can represent the superconducting and the superfluid films. For this model there is no phase transition showing long-range ordering. One example is the two-dimensional Coulomb gas model where the vortex-antivortex pairs, which are bound to each other at low temperature, increases in number as the temperature increases and become separated at the KT temperature that marks the phase transition.

It had been generally believed that no phase transition can exist for the $xy$ model when Kosterlitz *et al* (1973) showed that there is another kind of phase transition, arisen from the topological excitation of vortex-antivortex pairs instead of from the long-range ordering found in a spontaneous magnetisation. They consider the two-dimensional model of gas with charges $\pm q$ where the interaction potential is $U(|\mathbf{r}_i - \mathbf{r}_j|)$ is $-2q_i q_j \ln|(\mathbf{r}_i - \mathbf{r}_j)/\mathbf{r}_0| + 2\mu$ when $r > r_0$, and 0 when $r < r_0$. The problem is reduced to that of solving an equation of the form $\frac{(\mathrm{d}y}{\mathrm{d}x)=-e^{-xy}}$. The application mentioned there is in the $xy$ model of magnetism, the solid-liquid transition, and the neutral superfluid, but not in a superconductor and a Heisenberg ferromagnet.

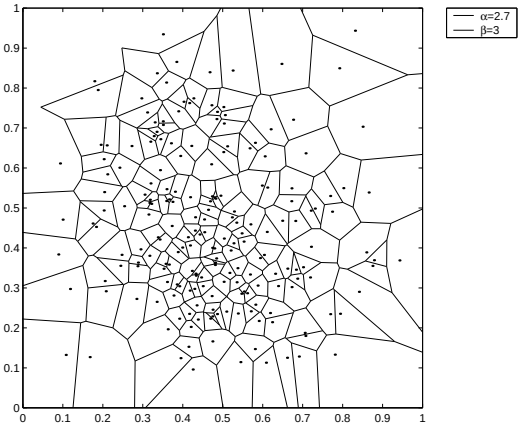
The frustrated $xy$ model, the hamiltonian of which is $H = J_0 \sum_{<i,j>} \cos(\theta_i - \theta_j - A_{ij})$, occurs when a magnetic field is applied perpendicular to the two-dimensional plane of the $xy$ model. The frustration parameter, $f = \Phi/Phi_0$, is a measure of the average external magnetic flux. When $f = 1/2$ the model is called the fully frustrated $xy$ model. The local chirality, $m(r_i) = \frac{1}{2\pi} \sum(\theta_i - \theta_j - A_{ij})$, which describes the property of the ground state, where it can either be $+1/ovr2$ or $-1/ovr2$. The network configuration at $\mathrm{T} < \mathrm{T}_c$ is that of a draught board, and has $\mathrm{Z}_2$ symmetry. This regularity is broken by the formation of domain walls in an Ising phase transition at $\mathrm{T}_c$.

Renormalisation group method has shown that there exist larger structures that behave like a smaller one. This means that the same structure can recur infinitely many times in infinitely many different scales, and that is the same idea that makes fractal geometry. This is why the study of cluster structure and the use of fractal dimension to characterise clusters becomes important (*cf* Stauffer and Aharony, 1985).

Percolation in Physics means Statistical Mechanics. Much of the contemporary vocabularies in the former has come from the lexicons of the latter from Ising Models to Renormalisation group and then to finite size scaling (*cf* Pathria, 1996).

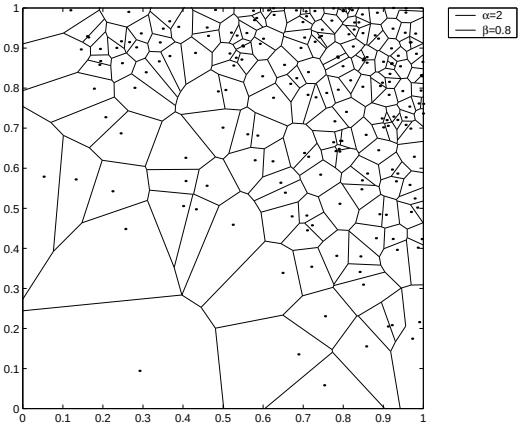
## § 1.9 Random processes

A synonym to *random* is *stochastic* (*cf* Miles, 1972). Any algorithm which employs a random element is called Monte Carlo. Random processes can have various types of distribution. The beta distribution has a probability density function $f_{X_{\alpha,\beta}}(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$, $0 \leq x \leq 1$, where $\alpha > 0$ and $\beta > 0$ are shape parameters, and $B(\alpha,\beta)$ is the beta function. There are three types of shape; the bridge shape has $\alpha > 1$ and $\beta > 1$, the J shape $\alpha \leq 1$ and $\beta \geq 1$, or $\alpha \geq 1$ and $\beta \leq 1$, and the U shape $\alpha < 1$ and $\beta < 1$.



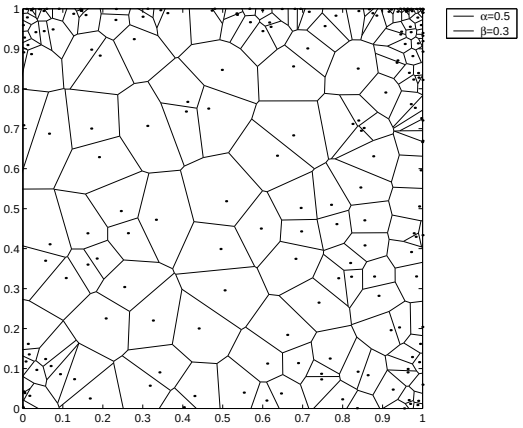

The 200 generators used in Figure 1.8 are randomly chosen with beta distribution with the shape parameters $\alpha = 2.7$ and $\beta = 3$, that is bridge shape.

**Figure 1.8** Voronoi graph with bridge-shaped beta distribution..




Both $x$ and $y$ in Figure 1.9 have J-shaped distribution with the shape parameters $\alpha = 2$ and $\beta = 0.8$. The density is unbounded at $x = 1$ and at $y = 1$ because $\beta < 0$ for both.

**Figure 1.9** Voronoi graph with J-shaped beta distribution..




The shape parameters in Figure 1.10 are $\alpha = 0.5$ and $\beta = 0.3$, that is U shape. The density is unbounded at $x = 0$, 1 and at $y = 0$, 1 because $\alpha$ is also less than zero.

**Figure 1.10** Voronoi graph with U-shaped Beta distribution..

The probability density function of some of the distributions of random numbers found in practice are shown in Table 1.9. The random variable of the F-distribution is $F_{n_1,n_2} = n_2 X/(n_1(1 - X))$, where $X$ is a beta variate with $\alpha = n_1/2$ and $\beta = n_2/2$. For the negative binomial distribution,

$p$ is a probability of success and is constant and $r$ the number of successes required before stopping. With the mean of $rq/p$, $q = 1 - p$, and the variance $rq/p^2$,it is used to model consecutive trials. For the noncentral$f$ distribution, $S_{n_1,\mu_1}$ and $S_{n_2,\mu_2}$ are chi-squared random variables which are independent and non-central, $S_{n,\mu} = \chi^2_{n-1} + (Z_n - \mu^{1/2})^2$. I has as its mean $\nu_2(\delta + \nu_1)/\nu(\nu_2 - 2)$, where $\nu > 2$, and as its variance $2(\nu_2/\nu_1)^2 \left[(\delta + \nu_1)^2 + (2\delta + \nu_1)(\nu_2 - 2)/(\nu_2 - 2)^2(\nu_2 - 4)\right]$, where $\nu_2 > 4$. This means that it is the general case of the $f$-distribution, which is the case where $\delta = 0$. The parameters $\nu_1$ and $\nu_2$ are degrees of freedom.

The noncentral $t$ distribution has $F(x|a,b)$ as the incomplete beta function with parameters $a$ and $b$, and $\nu$ the degrees of freedom. It is the generalisation of the student's $t$-distribution. It's mean is $\delta(\nu/2)^{1/2}\Gamma((\nu-1)/2)/\Gamma(\nu/2)$, where $\nu > 1$, and it's variance is $\nu/(\nu-2)(1+\delta^2) - \nu/2\delta^2\left[\Gamma((\nu-1)/2)/\Gamma(\nu/2)\right]^2$. The noncentrality parameter for all noncentral distributions, *viz.* the noncentral $f$-, $t$- and chi-square distributions is represented by $\delta$.

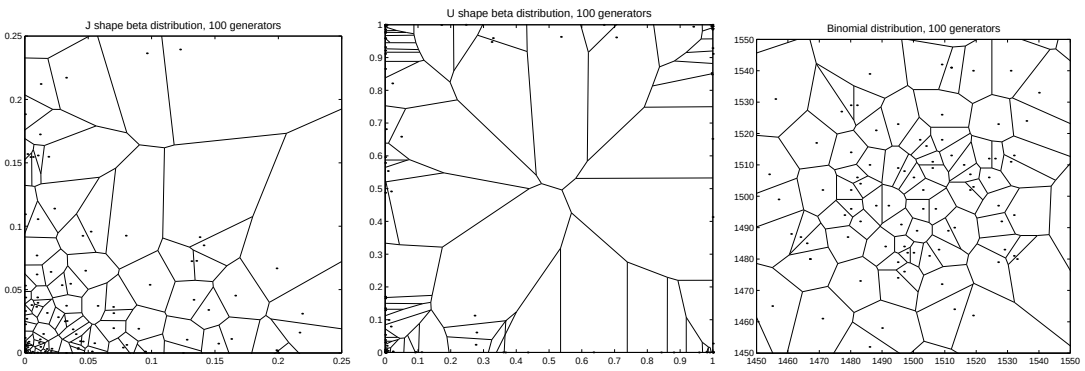
The mean of the normal distribution is $\mu$ and the variance $\sigma^2$. The standard normal distribution has $\mu = 0$ and $\sigma = 1$. The probability mass function of the Poisson distribution is sometimes written $y = f(x|\lambda) = \lambda^x/x!e^{-\lambda}\mathrm{I}_{0,1,\ldots}(x)$. It has a value when $x$ is a nonnegative integer. Otherwise the density function is zero.

The Rayleigh distribution has a parameter $b$. Its mean is $b\sqrt{x/2}$ and its variance $(4-x)b^2/2$. The uniform distribution has the mean $(a+b)/2$ and the variance $(b-a)^2/12$. For the standard uniform distribution $a = 0$ and $b = 1$. The discrete uniform has the mean $(N+1)/2$ and the variance $(N^2-1)/12$.

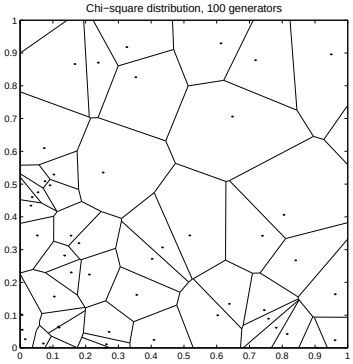
The Weibull distribution is sometimes written $y = f(x|a,b) = abx^{b-1}e^{-ax^b}\mathrm{I}_{0,\infty}(x)$. The Weibull distribution with a single parameter has $a = 1$. The three-parameter Weibull distribution has a p.d.f. $f_X(x) = c(x-a)^{c-1}e^{-((x-a)/b)^c}b^{-c}$ when $x \geq a$, otherwise $f_X(x) = 0$.. The mean is $a^{-(1/b)\Gamma(1+b^{-1})}$ and the variance is $a^{-2/b}\left[\Gamma(1+2b^{-1}) - \Gamma^2(1+b^{-1})\right]$.

| | |
|---|---|
| Beta | $f_{X_{\alpha,\beta}}(x) = x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha,\beta)$, $0 \leq x \leq 1$ |
| Binomial | $f_{X_{n,p}}(x) = {}^n\mathrm{C}_x p^x q^{n-x}$, $x = 0, 1, \ldots, n$, $0 \leq p \leq 1$ and $q = 1-p$ |
| Chi-square | $f_{\chi^2_n}(x) = e^{-x/2}x^{(n/2)-1}/2^{n/2}\Gamma(n/2)$, $0 \leq x \leq 1$ |
| Exponential | $y = f_{x\|\mu} = e^{-x/\mu}/\mu$ |
| F | $f_{F_{n_1,n_2}}(x) = n_1^{n_1/2}n_2^{n_2/2}x^{(n_1/2)-1}/B(n_1/2,n_2/2)(n_2+n_1x)^{(n_1+n_2)/2}$ |
| Lognormal | $y = f_{x\|\mu,\sigma} = e^{-(\ln(x)-\mu)^2/2\sigma^2}/x\sigma\sqrt{2x}$ |
| Negative Binomial | $y = f_{x\|r,p} = {}^{r+x-1}\mathrm{C}_x p^r q^x \mathrm{I}_{0,1,\ldots}(x)$, |
| Noncentral $f$ | $F_{n_1,n_2} = \sqrt{S_{n_1,\mu_1}/n_1}/\sqrt{S_{n_2,\mu_2}/n_2}$ |
| Noncentral $t$ | $\mathrm{P}(-t < x < t\|(\nu,\delta)) = \sum_{j=0}^{\infty}\left[(\delta^2/2)^j e^{\delta^2/2}/j!\right]\mathrm{I}\left(x^2/(\nu+x^2)\|1/2+j,\nu/2\right)$ |
| Noncentral Chi-square | $F(x\|\nu,\delta) = \sum_{j=0}^{\infty}\left[(\delta/2)^j e^{\delta/2}/j!\right]\mathrm{P}(\chi^2_{\nu+2j} \leq x)$ |
| Normal | $y = f(x\|\mu,\sigma) = e^{-(x-\mu)^2/2\sigma^2}/(\sigma\sqrt{2\pi})$, $\mu > 0$ |
| Poisson | $f_X(x) = \mu^x e^{-\mu}/x!$, $x \geq 0$ |
| Rayleigh | $y = f(x\|b) = (x/b^2)e^{(-x^2/2b^2)}$ |
| T | $y = f(x\|\nu) = \Gamma((\nu+1)/2)/\Gamma(\nu/2)\cdot 1/\sqrt{\nu x}\cdot 1/(1+x^2/\nu)^{\nu+1/2}$ |
| Uniform | $y = f(x\|a,b) = (1/(b-a))\mathrm{I}_{[a,b]}(x)$, $b > a$ |
| Discrete Uniform | $y = f(x\|N) = (1/N)\mathrm{I}(1,\ldots,N)(x)$ |
| Weibull | $f_X(x) = ax^{a-1}e^{-(x/b)^a}/b^a$, $x \geq 0$ and $a, b > 0$ |

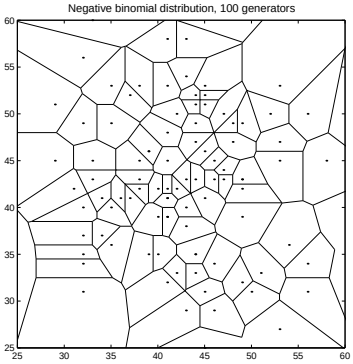
**Table 1.9** *Probability density functions.*


J shape beta distribution, 100 generators
U shape beta distribution, 100 generators
Binomial distribution, 100 generators

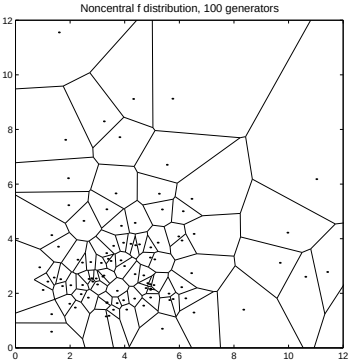
(a) (b) (c)


Chi−square distribution, 100 generators

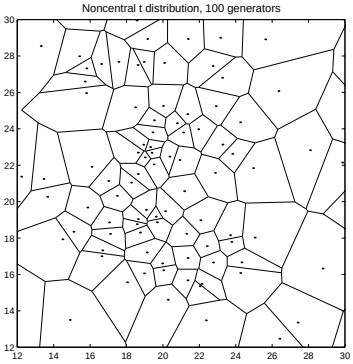
(d)


Negative binomial distribution, 100 generators

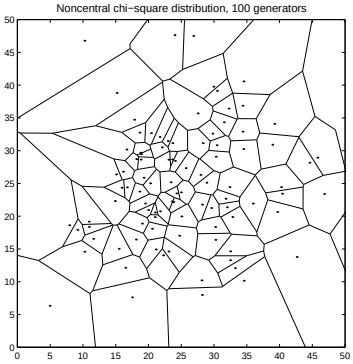
(e)


Noncentral f distribution, 100 generators

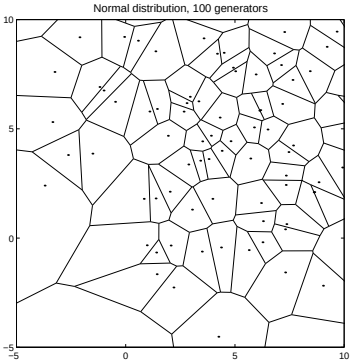
(f)


Noncentral t distribution, 100 generators

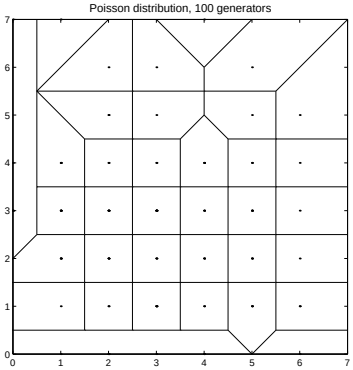
(g)


Noncentral chi−square distribution, 100 generators

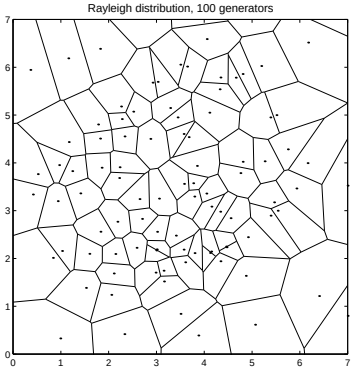
(h)


Normal distribution, 100 generators

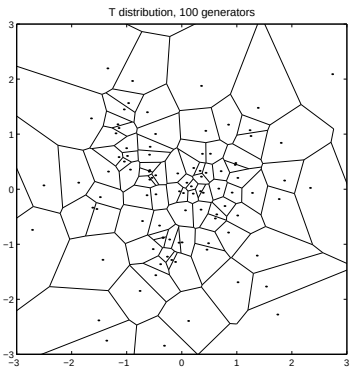
(i)


Poisson distribution, 100 generators

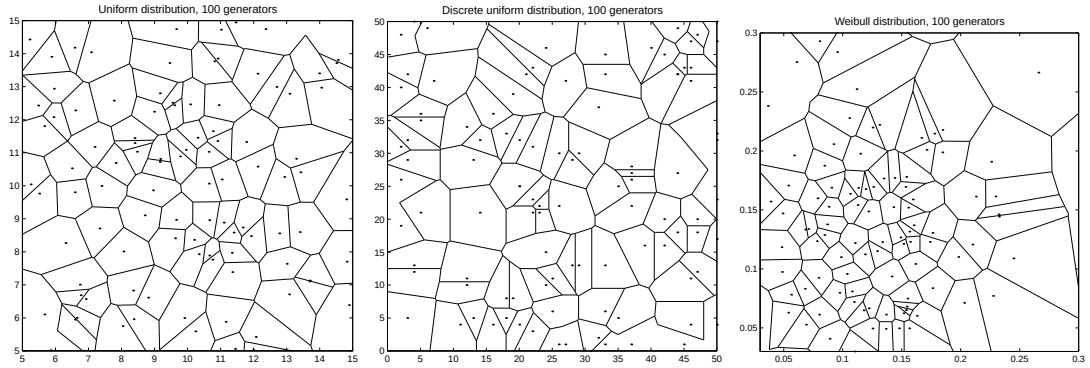
(j)


Rayleigh distribution, 100 generators

(k)


T distribution, 100 generators

(l)


Uniform distribution, 100 generators
Discrete uniform distribution, 100 generators
Weibull distribution, 100 generators

(m) (n) (o)

**Figure 1.11** *Voronoi graphs of various random p.d.f.'s, whose distributions are* (a) *J-shaped Beta,* $\alpha = 0.5$ *and* $\beta = 10$, *in both* $x$ *and* $y$, (b) *U-shaped Beta,* $\alpha = 0.1$ *and* $\beta = 0.2$, (c) *Binomial,* $n = 3{,}000$ *and* $p = 0.5$, (d) *Chi-squared,* $n = 1$, (e) *negative binomial,* $p = 0.7$ *and* $r = 100$, (f) *noncentral* $f$, $\nu_1 = 7$ *and* $\nu_1 = 12$, $\delta = 20$, (g) *noncentral* $t$, $\nu = 10$ *and* $\delta = 20$, (h) *noncentral chi-square,* $\nu = 7$ *and* $\delta = 20$, (i) *normal,* $\mu = 5$ *and* $\sigma = 4$, (j) *Poisson, 100 generators* $(x, y)$, $\lambda = 3$, (k) *Rayleigh,* $b = 3$, (l) $t$,$\nu = 7$, (m) *uniform,* $a = 15$ *and* $b = 5$, (n) *discrete uniform, from 100 points,* $N = 50$ *and* (o) *Weibull,* $a = 50$ *and* $b = 2$.

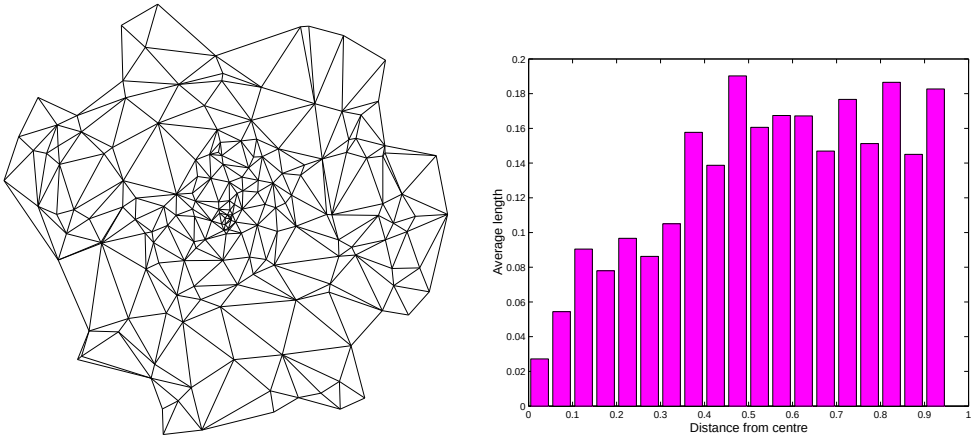
There are five methods for generation of Poisson processes, two of which are the time-scale transformation and the thinning algorithm. The first one changes a heterogeneous process $T_{(i)}$ in $[0, t_0]$ into a homogeneous process $\tau_{(i)}$ by a new time scale $\tau = \in t_0^T \lambda(u)du$, where $\lambda$ is the rate function. We have $\tau_{(i)} = \Delta(T_{(i)})$ and $T_{(i)} = \Delta^{-1}(\tau_{(i)})$, for $\tau_{(i)} \leq \Delta(t_0)$. If $\lambda(t) = e^{\alpha+\beta t}$, then $\Delta(T) = e^{\alpha}(e^{\beta T} - 1)/\beta$ and $T_{(i)} = (1/\beta)\ln(\beta\tau/e^{\alpha} + 1)$. The second one follows Algorithm 1.3 with the input $\lambda^*, \lambda(\cdot), t_0$.

**Algorithm 1.3** *Thinning algorithm.*

$T \leftarrow 0$;
**while** $T \geq t_0$ **do**
  **generate** $U$, $V \sim U(0,1)$;
  **while** $V > \lambda(T)/\lambda^*$ **do**
    $T \leftarrow T - \ln U/\lambda^*$;
  **endwhile**
**endwhile**

□

Figure 1.12 shows that the result from the point process in two dimensions involving two random variables, *i.e.* $r(\theta) = r\angle\theta$, is not homogeneous even though it may be on average isotropic with respect to the centre for a very large network. In Figure 1.13 where the domain is also circular but each point is simply a 2-d Poisson point process, the average distance is constant for a large system. Figure 1.14 is also 2-d Poisson point process, but the space here is square. In all of the figures, *i.e.* Figure 1.12–1.14, (a) and (b) start from 200-, whereas (c) and (d) from 1,000 point generators. Boundary effects are reduced by excluding those vertices and edges along the border.



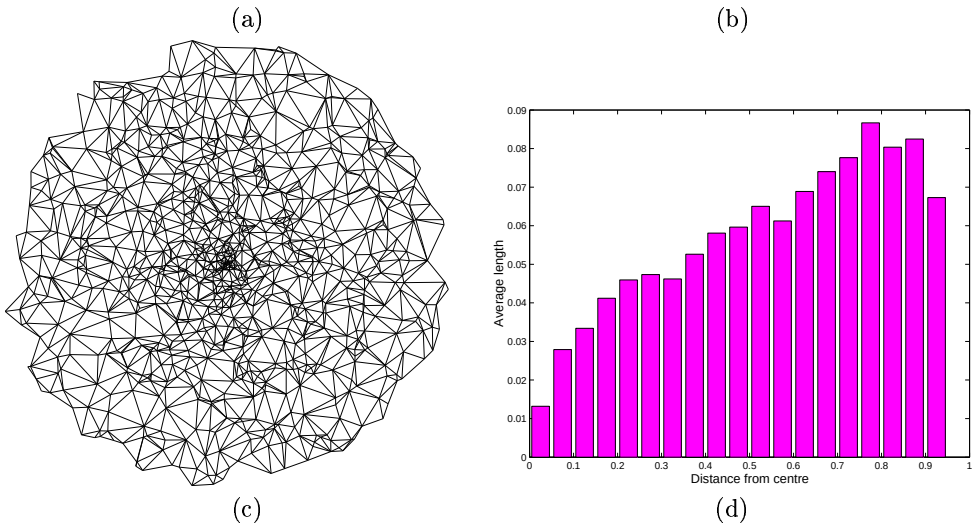



**Figure 1.12** *Point process $r\angle\theta$, where both $r$ and $\theta$ are independent random generators; (a) the Delaunay triangulation, 182 vertices shown, (b) average distance between neighbours vs distance from centre of the network, (c) another similar network, 947 vertices shown in total, (d) average edge length of the Delaunay triangulation.*

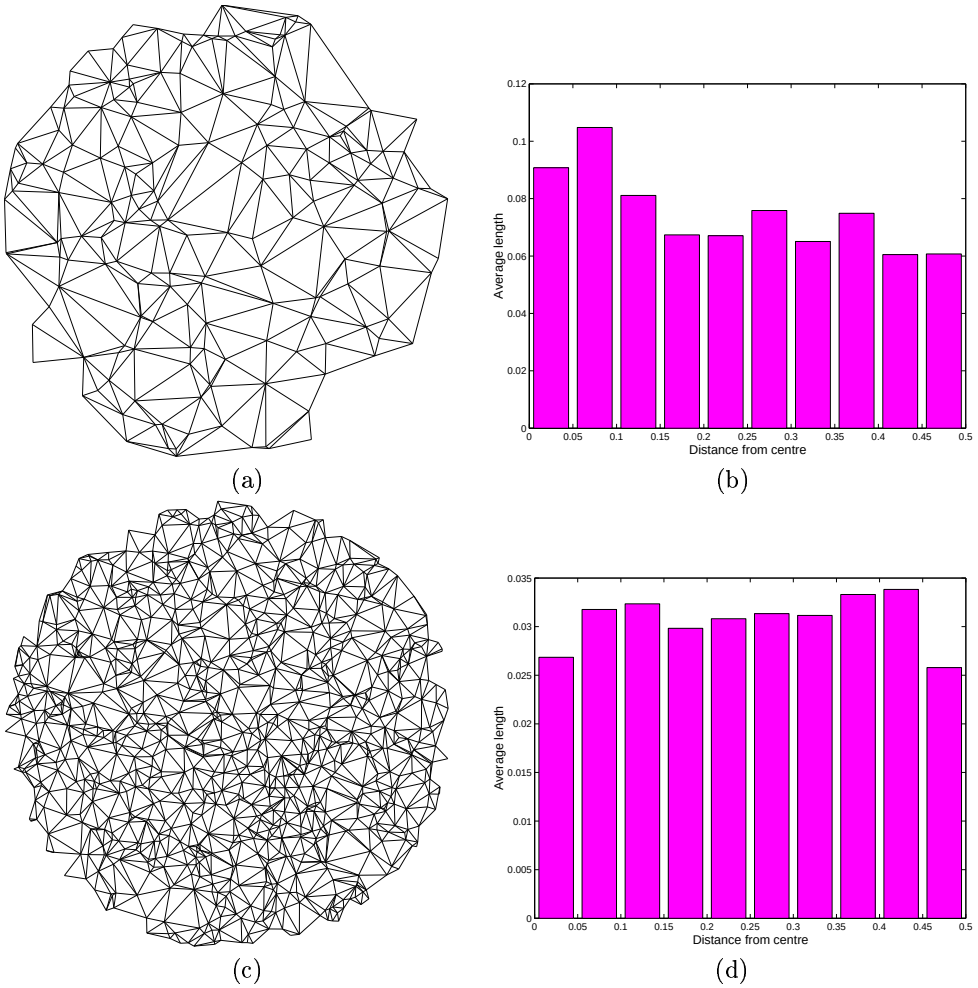



**Figure 1.13** *Point process $x, y$, where $x$ and $y$ are independent random generators in a circular domain; (a) the Delaunay triangulation with 184 vertices, (b) average edge length vs distance from centre of the network, (c) another similar network, 909 vertices shown in total, (d) average edge length of the Delaunay triangulation.*

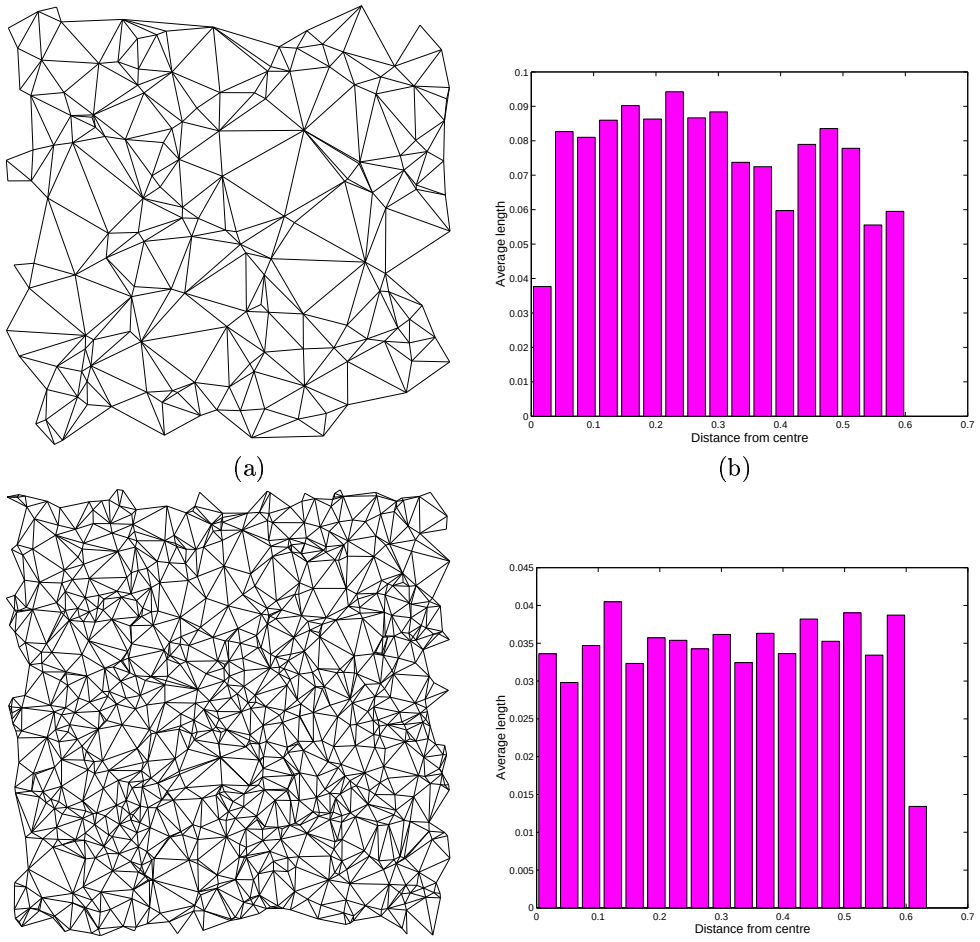



**Figure 1.14** *Point process $x, y$, where $x$ and $y$ are independent random generators within a square domain; (a) the Delaunay triangulation with 168 vertices, (b) average edge length vs distance from centre of the network, (c) another similar network, 816 vertices shown in total, (d) average edge length of the Delaunay triangulation vs the distance from centre.*

The codes used in carrying out these investigations are listed in § A.23. Figure 1.14 effectively tells us that the random point process $r\angle\theta$ gives particle distribution shaped like a bell with density decreasing away from the centre. One might almost say that this is similar to what a spiral galaxy looks like. There is still much room to explore what particle distribution the different topologies of random process give rise to, for example, 'What would the distribution in 2-d be of points that are generated from random points on a line swept in circle at a constant speed?' The algorithm is as simple as Algorithm 1.4, but the room for imagination and the scope of exploration are unlimited. There are only two variables, that is the nuclei positions $x$ and the triangulation edges $e$.

**Algorithm 1.4** *Triangulation edge length distribution.*

$x \leftarrow$ random process;
$e \leftarrow$ **find** triangulation of $x$;
$(x, e) \leftarrow (x - \partial x, e - \partial e)$;
**compare** $|e|$;

□

## § 1.10 Structures in nature

Prusinkiewicz and Lindenmayer (1990) model the structures in plants after having briefly discussed about the difference between the Chomsky grammars and the L-system, both of which being a mean for doing string rewriting, but the latter is based on the Turtle geometry which makes it convenient for the geometric rewriting of fractals. The states of a turtle consists of its position coordinates and the direction in which it is facing. Meinhardt (1995) models the patterns on sea shells by using mathematical based on the partial differential equations governing the system of activator, inhibitor, and substrate. Starting from a homogeneous initial condition, small deviations therein undergo a positive feedback and therefore increase. Activator catalyses both the production of itself and that of its inhibitor. The latter acts as a negative feedback which limits and makes the reaction local.

Random tissue in three dimensions has four edges, six faces and four cells meeting at each vertex. It is thus surrounded by four cell nuclei, as well as by six bonds forming a tetrahedral cage. The four edges meeting at a vertex resemble a caltrop, and similarly the lines from it to the four nuclei. If a vertex had more than four edges it would have been structurally unstable, because then it can be split into two normal vertices by an infinitesimal deformation. Continuous random networks, for example models of covalent glasses like vitreous silica, are excluded from this restricted class since theirs may be more than three non-planar faces to an edge even if they still have four edges to a vertex (Revier, 1982).

Unlike crystallography, the ideal random structure is by no means unique because it is the solution of a statistical problem. There are, however, certain geometrical and topological invariances, the most famous of which is possibly the Euler's theorem. In two dimensions this theorem states that $f - e + v = \chi$, where $\chi$ is an Euler-Poincaré characteristic and integer of order one, $\chi$ being for instance 1 and 2 respectively for plane and sphere; in three dimensions it is $f - e + v = 2$. The valence relations, $\sum nf_n = 2e = 3v$, hold for 2-d and 3-d alike.

Stumbling upon some observations which have gaven rise to Theorem's 3.3 and 3.4, I used both of the latter to explain by using Algorithm 3.2, page 100, the valence relations. Theorem 1.2 is also another product obtainable from applying both the Euler's theorem and the valence relations (*cf* Prause, 2000).

**Theorem 1.2.** *The average number of edges per polygon in a large pattern is six.:*

**Proof.:** From Euler's theorem, $f - e + v = 1$, and the valence relations, $\sum_n f_n = 2e = 3v$, it follows by applying the latter to the former that $f - \sum nf/2 + \sum nf/3 = 1$. Since $\bar{n}$ is the average number of edges per face, it follows that $f - \bar{n}f/2 + \bar{n}f/3 = 1$. Then $f(1 - n/6) = 1$, and consequently $1 - 1/f = n/6$. As the network becomes large, $f$ becomes infinite and as a result $\bar{n} = 6$. □

It is interesting to note that Theorem 1.2 posts no restrictions on whether the network is random or regular. The only assumption made is that three and only three edges meet at each vertex. One is almost tempted to say that, as the size of a network becomes infinite, nature somehow follows this theorem and make sure that each of the polygons has six edges on average.

When I translated the three papers by Voronoi (Tiyapan, 2001) I used the term *vertex* to mean a vertex of a specific polygon or polyhedron. For any vertex, I used the term *vertice*, and for more than one vertex vertices. It turns out that I am not the only one who concerns himself with the word. Moore and Angell (1993), for instance, use *apex* for a single corner, *vertex* to mean any point in a tessellation in two or three dimensions where its apices meet.

Rivulets flowing inside a pack bed have been studied by several authors (*cf* Porter, 1968). They are said to flow independently of each other, with no mixing among one another. Diffusion theory has been used to treat a random walk process.

The van der Waals force in combination with double layer repulsion play an important part in the study of filter and particle movements in porous media. Both are electrical forces and can be used to explain the particle capturing mechanism (*cf* Spielman, ).

Percolation is related to chemical engineering contexts (*cf* Mohanty *et al*, 1982). But the necessary background in stochastic processes for the purpose of simulation has already been described earlier, for example the modelling of colmatage, the retaining of particles suspended in a fluid flowing through a porous medium (Litwiniszyn, 1963 and 1967). In general, however, few authors in all engineering fields relate their works directly to the percolation theory. The majority of studies in this area are based on dynamics and fluid dynamics theories (*cf* Mulder and Gimbel, 1990; Kock and Judd, 1965).

The reason for the lack of percolation material in engineering literature may be because the percolation process generally works behind the scene and only shows itself as critical phenomena.

Most engineering studies are concerned with things under some operational condition, within the range of which percolation seems to be absent. By contrast, in Physics where extreme conditions are considered, there is an enormous and increasing amount of publications which are directly under the topic of percolation. But this by no means means that within a pacific range there is no place for percolation. In fact it is precisely this lack of mentioning in the literature that induces one to this kind of research. Because our idea of criticality is by tradition closely linked to the idea of time, percolation seems to be present only when there are instantaneous changes. But if in a steep s-curve we only rotate the axis clockwise by $\pi/2$ radian, such that to make the time axis vertical instead of horizontal, then we will see that in place of one critical point in the middle of the graph connecting two different levels, there are now two critical phenomena on both sides, one on each side, and in the middle a flat region where time hardly changes. Looking at it this way percolation seems to be a symmetry between time and space. In physical systems spaces percolates, but in the dual world where criticality is continuity it is the time instead which percolates.

Having said that, without rotating the time-space axes backwards and forwards too often one should still find ways to investigate what a pacified percolation does. In this regard, the study of economics seems to be an ideal place to start, if simply because one knows there exists such thing as hyperinflation but one never wants to study that when it happens. This automatically forces the researcher to find ways of doing researches which would not ruin his pocket or put his life in jeopardy. Another ground with a good prospect is in traffic congestion, eventhough its worst effect is not yet devastating, apart from what it sometimes does to the economy.

With these digressions in mind, if we now turn our thought at this point to our chemical engineering studies, we will not fail to see how ideal filtration fits the no-ruins requirement. For here we have a process which percolates routinely, often needs to be backflushed, but where the effect it produce is probably that of giving engineers a headache and, at its worst, putting a decent company out of business. But even with this advantage, I still believe that the study of filtration should not concentrate only on the fouling of filters, but should try to understand both the percolated and nonpercolated situations, preferably the latter for the lack of it in literature, and to connect what happens in a working filter to what happens, or does not happen, in a fouled one.

The revolutionising discovery made by F. August Kekulé (Kekulé, 1865, cited in Wotiz, 1993) that benzene has cyclic nature gave rise to the structural theory of organic chemistry.

## § 1.11 Computational geometry

The altitude lines of a triangle are concurrent. The bisectors of the angles of a triangle are concurrent. Ceva's theorem says that, all the three lines in a triangle which contain a vertex and a point on its opposite side are concurrent if and only if no two among them are parallel and the product of the three ratios of division of the sides made in one direction around the circumference of the triangle is one. In Gergonne's theorem, the three lines of a triangle which are made by the vertices and the points of tangency of the incircle on the side opposite to them are concurrent. The intersection between the three lines tangent to the circumcircle of a triangle and the sidelines opposite to them are collinear.

A point is an extreme point of a plane convex set $s$ unless it lies in a triangle which has vertices in $s$ but is no vertex of the triangle. A ray from inside a bounded convex figure intersects the boundary of the latter at exactly one point. Consecutive vertices of a convex polygon exist in sorted angular order about any interior point. A subfacet of a simple polytope is shared by two and only two facets. Two facets share a subfacet if and only if the latter is determined by $d-1$ vertices in their set; these two facets and the subfacet are called adjacent. A line segment defined by two points is an edge of the convex hull if and only if all other points of the set lie on, or to one side of it.

The diameter of a convex figure is the largest distance between parallel lines of support. The diameter of its convex hull determines the diameter of a set. Every vertex of the Voronoi graph is the intersection of three of its edges. Every nearest neighbour of a Voronoi polygon defines an edge.

VT and the triangulation of its nuclei are dual to each other. A Voronoi graph on $n$ points has at most $2n-5$ vertices and $3n-6$ edges. The convex hull of a Voronoi graph on $n$ can be found in linear time.

Modern programming philosophy puts much emphasise on modularity of a program and on information hiding of modules. Though undoubtedly information hiding can be good for the finished products, during the course of development it sometimes works against yourself when you try to pin-point an error in order to debug. Some modularisations are more about hierarchies than simplicity. Whenever this is the case, it is necessary to unconventionally seek to a simpler path.

The explicit equation in 2-d is $y = mx + c$ or $ax + by + c = 0$. Imposing the constraint $a^2 + b^2 = 1$, that is multiplying all the terms by $(a^2 + b^2)^{-1/2}$, puts the equation into the canonical or normalised form. This makes $a = \cos\alpha$, $b\cos\beta$ and $c = -r$, where $a$ and $b$ are directional cosines, *i.e.* the cosines of the angles that the normal line makes with the $x$ and $y$ axes respectively. Examples of possible conventions are to have a normal line point towards outside of the region, to have the line direction always to the right of the normal vector, or to keep $c$ positive always.

A parametric form of line equation in 2-d is by introducing a third variable $t$ and write the equations as $x = x_0 + ft$ and $y = y_0 + gt$, where $(x_0, y_0)$ is the point on the line corresponding to $t = 0$. A line through a point $p$ which makes angles $\alpha$ and $\beta$ with the $x$ and the $y$ axes respectively has the parametric equations $x = x_p + t\cos\alpha$ and $y = y_p t\cos\beta$. One convention is to vary $t$ from 0 to 1 over a line segment, another is to normalise it by multiplying its coefficient by $(f^2 + g^2)^{-1/2}$.

An implicit line equation $ax + by + c = 0$ can be turned into a parametric form as $x = -ac/(a^2 + b^2)^{1/2} + bt$ and $y = -bc/(a^2 + b^2)^{1/2} - at$. And a parametric line described by $x = x_0 + ft$ and $y = y_0 + gt$ is converted into the implicit form as $-gx + fy + (x_0 g - y_0 f) = 0$

The implicit plane equation in three dimensions is $ax + by + cz + d = 0$. The parameters can be found by using Cramer's rule, $a = \det(1, y_i, z_i)$, $b = \det(x_i, 1, z_i)$, $c = \det(x_i, y_i, 1)$ and $d = \det(x_i, y_i, z_i)$, which gives $a = y_1 z_{32} + y_2 z_{13} + y_3 z_{21}$, $b = z_1 x_{32} + z_2 x_{13} + z_3 x_{21}$, $c = x_1 y_{32} + x_2 y_{13} + x_3 y_{21}$ and $d = x_1(y_2 z_3 - y_3 z_2) + x_2(y_3 z_1 - y_1 z_3) + x_3(y_1 z_2 - y_2 z_1)$.

A normalised form has a constraint $a^2 + b^2 + c^2 = 1$. This amounts to multiplying its implicit equation by $(a^2 + b^2 + c^2)^{-1/2}$ to get $\alpha x + \beta y + \gamma z + \delta = 0$. Here $\alpha$, $\beta$ and $\gamma$ are cosines of the angles which the normal to the plane makes with the coordinate axes. The distance between two parallel normalised planes is $\delta_2 - \delta_1$. A normalised implicit plane equation can be used to represent a planar half-space by multiplying every tems by $-1$ and then assign a convention that the vector formed by the direction cosines always points towards the outside or the inside of the region.

The distance from a point to a plane, if the plane is $ax + by + cz + d = 0$ and the point is $(x_p, y_p, z_p)$, is $r = \left[(ax_p + by_p + cz_p + d)^2/(a^2 + b^2 + c^2)\right]^{1/2}$. The intersection of two planes, from the planes $a_1 x + b_1 y + c_1 z + d_1 = 0$ and $a_2 x + b_2 y + c_2 z + d_2 = 0$, is $x = x_0 + ft$, $y = y_0 + gt$ and $z = z_0 + ht$ where $f = \det(b_i, c_i)$, $g = \det(c_i, a_i)$ and $h = \det(a_i, b_i)$, $i = 1$ and 2.

The intersection of three planes is found by Algorithm 1.5, Here the minor matrices $a^{ij}$ is $\delta^{ab}_{23}$, $\delta^{ac}_{23}$, or $\delta^{bc}_{23}$ as the case may be.

**Algorithm 1.5** *Intersection among three planes.*

$\Delta \leftarrow \sum_{(a,b,c)}(-1)^{i+j} a_{ij} a^{ij}$;
**if** $|\Delta| < \epsilon$ **then**
  at least two of the planes are parallel;
**else**
  $x \leftarrow (b_1\delta^{dc}_{23} - d_1\delta^{bc}_{23} - c_1\delta^{db}_{23})/\Delta$;
  $y \leftarrow (d_1\delta^{ac}_{23} - a_1\delta^{dc}_{23} - c_1\delta^{ad}_{23})/\Delta$;
  $z \leftarrow (b_1\delta^{ad}_{23} + a_1\delta^{db}_{23} - d_1\delta^{ab}_{23})/\Delta$;
**endif** □

The intersection between a line and the plane $ax + by + cz + d = 0$ is $(x_1 + x_{12}a, y_1 + y_{12}a, z_1 + z_{12}a)$, where $a = -(ax_1 + by_1 + cz_1 + d)/(ax_{12} + by_{12} + cz_{12})$, $x_{12} = x_2 - x_1$ and similarly for $y_{12}$ and $z_{12}$.

The area of a circle is $\pi r^2$ and that of its segment is $\theta r^2/2$. A segment is its pie cut reaching its centre while a sector is a plane slice through the sphere. The area of a sector is this area subtracted by that of a triangle, or $r^2(\theta - \sin\theta)/2$. The centre of gravity or the centroid lies on the bisector of the central angle with the distance of $4r\sin(\theta/2)/3\theta$ for a sector and $4r\sin^3(\theta/2)/3(\theta - \sin\theta)$ for a segment.

The volume of a pyramid is $Ah/3$, where $A$ is the area of base and $h$ is the height of the pyramid. The volume of a sphere is $4\pi r^3/3$, and the distance from its centroid to the sphere centre is $(3r/4)\left[\sin^4(\theta/2)/(2 - 3\cos(\theta/2) + \cos^3(\theta/2))\right]$ That of a sector of a sphere is $(\pi r^3/3)(2 - 3\cos(\theta/2) + \cos^3(\theta/2))$ The volume of a tetrahedron is $V = (1/6)\det(x_{12}, x_{13}, x_{14}; y_{12}, y_{13}, y_{14}; z_{12}, z_{13}, z_{14})$, where $x_{ij} = x_j - x_i$,or $V = (1/6)\det(x_i, y_i, z_i, 1)$. The former is limitted to the case of three dimensions, and is in fact $V = (1/6)(a \times b) \cdot c$, where $a$, $b$ and $c$ are respectively the lines from O to A, B and C in a tetrahedron OABC.

Generalising the latter to higher dimensions, we have the volume of a $d$-dimensional simplex $V = (1/d!)\det(x_{ij}, 1)$, where $x_{ij}$ is now $(x_j)_i$, $1 \le i \le (d + 1)$ and $1 \le j \le d$. Here the information that I have found in existing literature seems to be wrong, some lists the multiplying factor as $1/d$,

some simply uses 1/6 throughout all ($d \geq 3$)! I had arrived by myself at the value which I am using; § 3.13 mentions this in more detail.

Some of of the algorithms found in literature are the following. The algorithm to find whether a point is inside a polygon, Algorithm 1.6. The arbitrary line $l$ here, which is taken for simplicity to be horizontal, passes through $z$.

**Algorithm 1.6** *Point inside a polygon.*

$r \leftarrow 0$;
**for** $i = 1$ to $n$ **do**
  **if** edge $i$ and $l$ not parallel **then**
    **if** $i$ intersects $l$ to the left of $z$ at any point except its lower extreme **then**
      $r \leftarrow r + 1$;
    **endif**
  **endif**
  **if** $r$ odd **then**
    $z$ is internal to $p$;
  **else**
    $z$ is external;
  **endif**
**endfor** □

To find the inclusion in a convex polygon, $q \in p$ being a known fixed point within the polygon, find the wedge in which $z$ lies by doing a binary search and test whether $\angle(zqp_{i+1})$ is a right turn- while $\angle(zqp_i)$ a left turn angle. If $\angle(p_i p_{i+1} z)$ is a left turn angle, then $z$ is inside $p$.

The Euclidean minimum spanning tree may be obtained by Algorithm 1.7.

**Algorithm 1.7** *Euclidean minimum spanning tree.*

$f \leftarrow 0$;
**for** $i$ from 1 to $n$ **do**
  $s(p_i) \leftarrow 0$;
  $f \leftarrow p_i$;
**endfor**
**while** $f$ contains more than one number **do**
  $t \leftarrow f$;
  **if** (s(t)=j) **then**
    clean up;
    $j \leftarrow j + 1$;
  **endif**
$(u, v) \leftarrow$ shortest unselected edge incident on $t$, $u \in t$;
$t' \leftarrow$ tree in $f$ containing $v$;
$t'' \leftarrow$ **merge** $(t, t')$];
**delete** $(t')$ from $f$;
$s(t'') \leftarrow \min(s(t), s(t')) + 1$;
$f \leftarrow t''$
**endwhile** □

To rotate $v = (v_1, v_2)^{\mathrm{T}}$ to $v' = (v_1', v_2')^{\mathrm{T}}$, use $v^{\mathrm{T}} = Av$ where $A = [\cos\theta, -\sin\theta; \sin\theta, \cos\theta]$ is the transformation matrix and $\theta$ is the anti-clockwise angle of the rotation.

To rotate a general line in three dimensions by $\theta$ around an arbitrary axis, the transformation matrix becomes $A = T^{-1}RT$, where

$$R(\theta) = R_x^{-1}(\alpha)R_y^{-1}(\beta)R_z(\theta)R_y(\beta)R_x(\alpha)T = R_x(-\alpha)R_y(-\beta)R_z(\theta)R_y(\beta)R_x(\alpha).$$

The translation $T$ translates one point of the line to the origin and $R_x$ rotates around the $x$-axis by $\alpha$ which puts $u$ on to the $xz$ axis, $R_y$ around $y$-axis by $\beta$ and puts $u'$ on to the $z$ axis, and $R_z$ around $z$-axis. Since the vector $v$ is $(x_{12}, y_{12}, z_{12})$, we have the $a$, $b$, $c$ and the unit vector $u = (a, b, c)/|v| = (x_{12}, y_{12}, z_{12})/|v|$. Then it follows that $\cos\alpha = c/d$, $\sin\alpha = b/d$, $\cos\beta = d$ and $\sin\beta = -a$. All of these can perhaps be summarised as a linear procedure in Algorithm 1.8.

**Algorithm 1.8** *Rotation in three dimensions*

$v \leftarrow (x_{12}, y_{12}, z_{12})$;
$m \leftarrow |v|$; $a \leftarrow x_{12}/m$;

$b \leftarrow y_{12}/m$;
$c \leftarrow z_{12}/m$;
$d \leftarrow (b^2 + c^2)^{1/2}$;
$r_{xx} = c/d$;
$r_{xy} \leftarrow b/d$;
$r_{yx} \leftarrow d$;
$r_{yy} \leftarrow -a$;
$T \leftarrow [\mathrm{I}(3), -(x_1, y_1, z_1)^{\mathrm{T}}; 0(3)^{\mathrm{T}}, 1]$;
$R_x \leftarrow [1, 0(3)^{\mathrm{T}}; 0, r_{xx}, -r_{xy}, 0; 0, r_{xy}, r_{xx}, 0; 0(3)^{\mathrm{T}}, 1]$;
$R_y \leftarrow [r_{yx}, 0, -r_{yy}, 0; 0, 1, 0, 0; -r_{yy}, 0, r_{yx}, 0; 0(3)^{\mathrm{T}}, 1]$;
$R_z \leftarrow [[\cos\theta, -\sin\theta; \sin\theta, \cos\theta], 0(2,2); 0(2,2), \mathrm{I}(2)]$;
$R \leftarrow T^{-1} R_y^{-1} R_z R_y R_x$; $v' = Rv$. □

Also in three dimensions, the rotation around the $x$-axis is

$$R_x = [\{r_{11} = 1, m_{23} = [\cos\theta, -\sin\theta; \sin\theta, \cos\theta]\}],$$

around $y$-axis is

$$R_y = [\{r_{22} = 1, m_{13} = [\cos\theta, \sin\theta; -\sin\theta, \cos\theta]\}]$$

and around $z$-axis is

$$R_z = [\{r_{33} = 1, m_{12} = [\cos\theta, -\sin\theta; \sin\theta, \cos\theta]\}].$$

The minor containing parts of the $i^{\mathrm{th}}$ and $j^{\mathrm{th}}$ rows and columns is $m_{ij}$.

The right hand coordinate system is where a 90° rotation around the $x$-, $y$- and $z$-axis bring respectively the $y$- to $z$-, $z$- to $x$- and $x$- to $y$-axis. Scaling and translating a vector $v$ in three dimensions amounts to calculating $[v'; w] = A[v; w]$, where $A$ is respectively $[\mathrm{I}(3)s, 0(3); 0(3)^{\mathrm{T}}, 1]$ and $[\mathrm{I}(3), \Delta v; 0(3)^{\mathrm{T}}, 1]$, $\Delta v = (\Delta x, \Delta y, \Delta z)^{\mathrm{T}}$.

A quaternion can be described as a pair $(s, v)$ of a scalar $s$ and a vector $v = (a, b, c)$. The rotation by $\theta$ around an axis in the direction of a unit vector $u$ is then the quaternion $(\cos\theta/2, u\sin\theta/2)$. Let $q = (s, v)$. Then $q^{-1} = (s, -v)$. The multiplication of quaternions is $q_1 q_2 = (s_1, v_1) \cdot (s_2, v_2) = (s_1 s_2 - v_1 \cdot v_2, s_1 v_2 + s_2 v_1 + v_1 \times v_2)$, where the cross product is described in minors as $v_1 \times v_2 = [\delta^{yz}; -\delta^{xz}; \delta^{xy}]$.

Let $q$ represent a rotation. Then a vector $p$ is rotated to $p'$ by $P' = qPq^{-1}$, where $P$ and $P'$ are respectively $(0, p)$ and $(0, p')$. In simplified words, this means $p' = s^2 p + (p \cdot v)v + 2s(v \times p) + v \times (v \times p)$. Then we have the transformation matrix for the general rotation around $u$ in three dimeneions,

$$R_u(\theta) = \begin{bmatrix} (1 - 2b^2 - 2c^2) & (2ab - 2sc) & (2ac + 2sb) \\ (2ab + 2sc) & (1 - 2a^2 - 2c^2) & (2bc - 2sa) \\ (2ac - 2sb) & (2bc + 2sa) & (1 - 2a^2 - 2b^2) \end{bmatrix}$$

where $s = \cos\theta/2$ and $v = (a, b, c) = u\sin\theta/2$. Furthermore, if $q_1$ is a rotation by $\theta_1$ around $v_1$ and likewise $q_2$ by $\theta_2$ around $v_2$, then $q_3 = q_2 q_1$ is a rotation by $\theta_3 = 2\cos^{-1} s_3$ around $v_3$ such that $\sin\theta_3 \geq 0$.

Quaternion is an extension of complex number to higher dimensions where there are three imaginary parts instead of one. It is defined as $q = s + ia + jb + kc$, where $a$, $b$, $c$ and $s$ are real numbers, $i^2 = j^2 = k^2 = -1$ and $ij = -ji = k$.

Let a plane be described by $(v - p) \cdot n = 0$, where $p$ is a point on-, and $n$ a perpendicular to the plane. This means that, for all points $v$ lying in the plane $(p, n)$, $(v - p) \cdot n = 0$. If the plane $(p, n)$ is transformed into $(Ap, m)$, then $Av$ lies in $(Ap, m)$ and consequently $A^{\mathrm{T}} m = n$ or $m = (A^{\mathrm{T}})^{-1} n$.

A vector normal to the surface remains normal if it is transformed by $(A^{\mathrm{T}})^{-1}$. When a transformation matrix $A$ has the property $A = (A^{\mathrm{T}})^{-1}$, for example rotation, all surface normals remain normal. But in general this equality does not hold, so we have for instance the non-uniform scaling where these normals cease to be normal.

To test for the intersection between a ray and a triangle using Plücker's coordinates is described in Algorithm 1.9.

**Algorithm 1.9** *Ray and triangle intersection.*

**find** Plücker's coordinates for vertices and the ray;
**test** ray against each of the edges;
**if** ray hits an edge, passes all of them clockwisely or all of them counter-clockwisely **then**
ray intersects the triangle;

**else**
ray and triangle intersect not;
**endif** □

If these vertices are $v_1$, $v_2$ and $v_3$, and the ray is $r = r_{12} = r_2 - r_1$, where $r_1$ and $r_2$ are any two points on the ray, then Plücker's coordinates for the vector $v_{12}$ are $(u, v) = (v_2 - v_1, v_2 \times v_1)$, and similarly for $v_{23}$, $v_{31}$ and $r_{12}$. For the test between the ray and each edge, find $c = u_r \cdot v_i + v_r \cdot u_i$. Then the ray $r$ counter-clockwisely passes the edge $i$, hits it or clockwisely passes it respectively as $c < 0$, $c = 0$ or $c > 0$.

A 3-d line can be represented by the six numbers that come with coordinates of two distinct points, or by the eight numbers that come with the coordinates of two distinct planes. Plücker's coordinates, however, provides a mean which suits geometrical computation better than both of these. It redefines the coordinates as $u = p - q$ and $v = p \times q$, where $p$ and $q$ are two points on a line, neither quantity of which depends on $p$ or $q$.

Let a tetrahedron has its vertices at $a_i$, $i = 1$ to 4. Then the centre of its circumsphere is at $a_1 + \delta$ and its corresponding radius $r = |\delta|$, where

$$\delta = \left[|a_{12}|^2(a_{13} \times a_{14}) + |a_{13}|^2(a_{14} \times a_{12}) + |a_{14}|^2(a_{12} \times a_{13})\right] / 2|a_{12}^{\mathrm{T}}; a_{13}^{\mathrm{T}}; a_{14}^{\mathrm{T}}|.$$

A triangle $\triangle p_1 p_2 p_3$, where $p_i = (x_i, y_i)$ and $i = 1$ to 3, has an area $A = |x, y, 1(3)|$, which is positive if and only if $\triangle p_1 p_2 p_3$ forms a counter clockwise cycle or $\angle p_1 p_2 p_3$ is left-turned. Or equivalently the area is $A = \det(x_{12}, x_{13}; y_{12}, y_{13})$, which is positive if the points are in anti-clockwise order of the indices and negative otherwise. Or the area is $A = [s(s - d_1)(s - d_2)(s - d_3)]^{1/2}$, where the the semi perimetre $s$ is $s = \sum_i d_i/2$, where $d_i$, $i = 1$ to 3, are the lengths of the three sides of the triangle.

For a convex polygon, the area can be found by adding together the $n$ triangles formed by any two adjacent vertices and one fixed point within the polygon. Here $n$ is the number of vertices it contains. Or it can be found by adding the $(n - 2)$ triangles formed by one fixed vertex and any two adjacent vertices of those remaining. Another way of finding the area is $A = (1/2)\sum_{i=0}^{n-1}(x_i y_{i+1} - y_i x_{i+1})$, where $(x_i, y_i)$ are vertices. Rearranging to make it faster and more accurate, $A = (1/2)\sum_{i=0}^{n-1}((x_i + x_{i+1})(y_{i+1} - y_i))$. If the dimension of the polygon is higher than two, $A = (1/2)\left|N \cdot \sum_{i=0}^{n-1}(v_i \times v_{i+1})\right|$, where $N$ is a unit vector normal to the plane. The area of a polygon can also be computed, without loosing generality, by subtracting the area under its lower edges by that of its upper edges.

A vector normal to a plane is simply $n = v_{12} \times v_{13}$.

The ordering of vertices on a face of a polygon is done for the purpose of drawing it or for finding vertice pairs which form edges. This can be done in two ways. One is to get inside the polygon and look at all the vertices around comparing their angles relave to one another. Another one is to look at the polygon from a distance and compare their angles as before, as well as their distance from the viewing point. The angle is $\theta = \arccos[(v_1 \cdot v_2)/(|v_1||v_2|)]$, where $v_1$ and $v_2$ are vectors to vertices from the distant point, and $\theta$ the angle between them.

To find the convex hull in three dimensions, one may use Algorithm 1.10.

**Algorithm 1.10** *Convex hull in three dimensions.*

**sort** $s$ by $x_1$ such that $x_i(p_i) < x_i(p_j)$ if and only if $i < j$;
**if** $|s \leq k|$ **then**
**construct** $c(s)$;
**else**
$s_1 \leftarrow \{p_1, \ldots, p_{\lfloor n/2 \rfloor}\}$;
$s_2 \leftarrow \{p_{\lfloor n/2 \rfloor}, \ldots, p_n\}$;
$p_1 \leftarrow c(s_1)$;
$p_2 \leftarrow c(s_2)$;
$p \leftarrow$ **merge** $p_1$ and $p_2$;
**endif** □

Here $c(s)$ is the convex hull of $s$. Two convex hulls are merged with each other by first constructing a cylindrical triangulation T which supports $p_1$ and $p_2$ along two circuits $e_1$ and $e_2$ respectively, then remove from both $p_1$ and $p_2$ the portions which have been obscured by T.

The following Jarvis's march algorithm, Algorithm 1.11, finds a convex hull in two dimensions.

**Algorithm 1.11** *Convex hull in two dimensions*

$p_1 \leftarrow$ the lowest point in $s$;
$q_1 \leftarrow$ the highest point in $s$;
**while** next point $\neq q_1$ **do**
**find** $p_i \in s$, $i = 2, 3, \ldots$, with an increasing order of the polar angles
with respect to $p_1$;
**endwhile**
**while** next point $\neq p_1$ **do**
**find** $q_i \in s$, $i = 2, 3, \ldots$, **with an increasing order of the polar angles with respect to** $q_1$ **and the negative** $x$ **axis;**
**endwhile** □

The polar angle is an angle with respect to the positive $x$-axis. The lowest and the highest points are on the convex hull.

Algorithm 1.12 is the quick hull algorithm.

**Algorithm 1.12** *Quick hull algorithm.*

$l \leftarrow (x_0, y_0)$;
$r \leftarrow (x_0, y_0 + \epsilon)$;
**if** $s = \{l, r\}$ **then**
**return** $(l, r)$;
**else**
**find** $k \in s$ that gives $\max A_{\triangle klr}$ or ($\max A_{\triangle klr}$ and $\max \angle klr$);
$s^1 \leftarrow p \in s$, such that $p$ is on the left of $\overrightarrow{lh}$;
$s^2 \leftarrow q \in s$, such that $q$ is on the left of $\overrightarrow{hr}$;
$\{h\} \leftarrow (s^1; l, h)$;
$\{h\} \leftarrow (s^2; h, r) - h$;
**endif** □

The convex hull is $\{h\}$. The points $l$ and $r$ are with respectively the smallest and the largest abcissa. In other words they are the left-most and the right-most points. And $k$ is the furthest point with respect to $l$ and $r$.

Let $p = \{p_1, p_2, \ldots, p_n\}$ be a set of $n$ generators in $\nu$-dimensional space, the coordinates of which are $(x_{ij})$, $i = 1$ to $d$ and $j = 1$ to $n$. Then the Delaunay tessellation in $d$ dimensions which spans $p$ is generated by,

**for** $j = 1$ to $n$ **do**
$\{q\} \leftarrow q_j = (x_{ij}, \sum_j x_{ij}^2)$;
**endfor**
$h \leftarrow c(q)$;
**project** all the lower $n$-faces of $c(q)$ parallel to the $d^{\text{th}}$ axis on to the original $n$-d space;
□

Okabe *et al* (1992) give a good review of algorithms for generating VT's. Given the set of generator points $\{p_i\}$, $i = 1$ to $n$, a brute force albeit simple method generates for all $i$ and $j$ from 1 to $n$ the $(n-1)$ half planes $h(p_i, p_j)$, $1 \leq j \leq n$, $i \neq j$, and then proceeds to construct all $\mathcal{V}(p_i)$ of the VT from their common intersections.

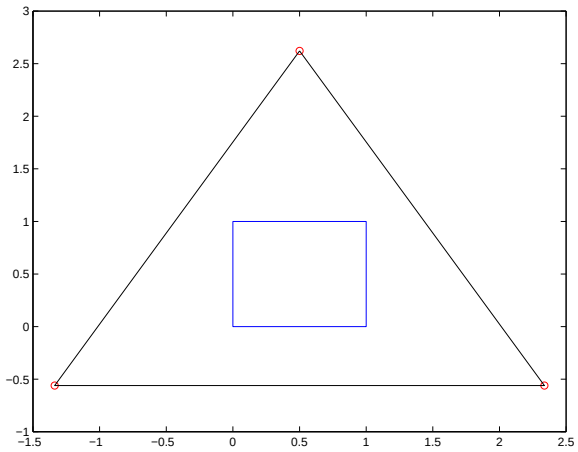
On the other hand, the following Algorithm 1.13 is the quaternary incremental method whose inputs comprise the $(n-3)$ generators $p_i$, $i = 4$ to $n$, where all the $p_i$ are in $s = \{(x, y) | 0 \leq x, y \leq 1\}$, and three additional generators $p_1 = (0.5, 0.5(1 + 3\sqrt{2}))$, $p_3 = (0.25(2 - 3\sqrt{6}), 0.25(2 - 3\sqrt{2}))$ and $p_3 = (0.25(2 + 3\sqrt{6}), 0.25(2 - 3\sqrt{2}))$.

**Algorithm 1.13** *Quaternary incremental method.*

$k \leftarrow \min(k_i)$ such that $k_i \in \mathrm{I}^+$ and $n < 4^{k_i}$;
$s_{ij} \leftarrow (i - 1, i/2^k, j - 1, j/2^k)$;
**construct** a quaternary tree;
**scan** leave buckets from left to right, top to bottom, **put** generators in buckets;
$\{p_i\}$ **do** quaternary reordering on the generators;
$\mathcal{V} \leftarrow$ **construct** Voronoi for $p_1$, $p_2$ and $p_3$;
**for** $i =$ **4 to** $n$ **do**
repeat $\leftarrow 1$;
**while** repeat **do**
**find** $p_\alpha$ such that $\mathrm{d}(p_\alpha, p_\ell) = \min_j \mathrm{d}(p_j, p_\ell)$;

**if** $\mathrm{d}(p_m, p_\ell) < \mathrm{d}(p_\alpha, p_\ell)$ **then**
repeat ¬**repeat;**
**elseif** $p_m \leftarrow p_\alpha$ **do** nothing;
**else**
repeat ¬**repeat;**
**endif**
**endwhile**
$\{w_{i1}, w_{i2}\} \leftarrow$ the intersections between the perpendicular bisector of $p_i p_l$ and $\Omega(\mathcal{V}(p_i))$, $1 \leq i \leq \ell - 1$;
$\Omega \leftarrow$ **construct** the boundary of $\mathcal{V}(p_i)$ formed by these $\overline{w_{i1} w_{i2}}$;
$\{\mathcal{V}_\ell\} \leftarrow (\{\mathcal{V}_{\ell-1}\} \bigcup \Omega) - \{\mathcal{V} | \mathcal{V} \in \{\mathcal{V}_{\ell-1}\}, \mathcal{V}$ **is within** $\Omega\}$;
**endfor**
$\{\mathcal{V}\} \leftarrow \{\mathcal{V}_n\}$;

□




The construction of $s_{ij}$ is such that $s_{ij} = (i-1, i/2^k] \times (j-1, j/2^k]$, where $i, j \in \mathrm{I}^+$, $i$ and $j$ are 1, 2, ... up to $2^k$. The nearest neighbour search finds from $\ell$ generators, $p_1, \ldots, p_\ell$, and $\{\mathcal{V}_{\ell-1}\}$, the generator point $p_m$ such that $\mathrm{d}(p_m, p_\ell) < \mathrm{d}(p_m, p_i)$ for all $i = 1$ to $\ell$, $i \neq m$ and $i \neq \ell$; $p_j$ are all generators adjacent to $\mathcal{V}(p_i)$. The boundary growing procedure gives the sequence of boundaries $\{\Omega\}$. The additional generators mentioned in the procedure is graphically shown in Figure 1.15.

**Figure 1.15** Additional generators for the incremental method.

If we draw a horizontal line through each point in two dimensions, or a horizontal plane each point in three dimensions, we divide the space into slabs the line segments within which do not intersect one another. In three dimensions, and for a polygonal model of porous media, we can for instance divide the model into such slabs, then find the effective cross sectional area of each slab, and then determine where the bottle neck to the flow occurs within the media. Whether this would produce the correct determination of the pressure of flow across the material is another matter because flows through porous media may be governed by the combination of the various tortuous paths through the pores, the interrelationship of which can be complicated.

These slabs provide another method in finding the area, or in three dimensions the volume, of each cell of the tessellation. Here the cell is divided into slabs, and then the area or volume of each section calculated.

In three dimensions, rotation around the $x$-axis is done by $\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$, around $y$-axis by $\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$ and around $z$-axis by $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

### § 1.12 Geometric algorithms

The program `findfarea.m` in § A.24 finds the area and plane parameters of a face from a matrix containing the list of the coordinates of the ordered vertices. Its synopsis is `(a,p)=findfarea(v)`, where `a` is the face area, `p` the list of the plane parameters and `v` the coordinate matrix.

Real problems are algebraical or analytical whereas computer simulations are arithmetical and numerical (*cf* Knuth, 1997). In between these two lie computer programs. Therefore the latter are mapping from the analytical to the arithmetical world. There are different ways to solve a problem analytically, and there are different ways numerically. Therefore numerical study by simulation is an endless persuit. Several programs listed in § Ado the same job, but each one does it differently.

Hamiltonian's are essentially a function that maps $N$ bodies pairwisely to account for all pairs of their mutual interactions. They occur in most fields where there are particles interacting with each other. In quantum mechanics, for example, the time independent Schrödinger equation for a system of $N$ particles interacting via the Coulomb interaction is $H\Psi = E\Psi$, where the Hamiltonian is

$$H = \sum_{i=1}^{N}\left(-\frac{\hbar}{2m_i}\nabla_i^2\right) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{z_i z_j}{4\pi\epsilon_0|r_i - r_j|}, \qquad (\mathbf{8})_{\mathrm{i}}$$

where $\Psi$ is the $N$-body wavefunction, $z$ the charges of the individual particles and $E$ the energy of either the ground or an excited state of the system. Similarly from Rushbrooke and Morgan (1961), using their notations, the Hamiltonian of the Ising problem is

$$\mathcal{H} = -2J\sum_{\langle i,j\rangle} s_3^{(i)}s_3^{(j)} - g\beta H\sum_{(i)} s_3^{(i)}, \qquad (\mathbf{9})_{\mathrm{i}}$$

where $\beta$ is the Bohr magneton, $g$ the gyromagnetic ratio and $J$ the magnitude of the exchange interaction.

When simulating such systems, the number of pairwise summation terms can be reduced by half because they represents a symmetric matrix. The total number of terms is thus reduced from $N(N-1)$ to $N(N-1)/2$ (*cf* Wray *et al*, 1983). Because $i \neq j$, all the diagonal components of the matrix are excluded, which makes the number of pairs $n^2 - n = n(n-1)$. In a program, this is equivalent to two $if$ statements, one embedded within the other in the form $i(j(\cdot))$, where the index $i$ runs from 1 to $(n-1)$ and $j$ from $(i+1)$ to $n$; this I discovered by myself from experience, as can be seen by comparing the present work with my previous work (Tiyapan, 1995, KNT2(ii)).

# 2

### § 2. Division of space

A stochastic system which is stochastically invariant under arbitrary translation, that is under the transformation $x \to a + x$, is homogeneous. Homogeneity lets us use the ergodic theory, namely $\dot{F}_{Z,q}(z) \to \dot{F}_Z(z)$ with probability one as $q$ approaches infinity for all $z$ in $Z$, in other words the empiric- implies the ergodic distribution function of the characteristic $Z$ for the cell $C$ in $Q(q)$. Empirical moments, $\dot{E}_q(z) = \int z\mathrm{d}\dot{F}_{Z,q}$, almost surely converge to the corresponding ergodic moments, $E(z) = \int z\mathrm{d}F_Z(z)$. The study of random division of space aims at defining classes of parameter dependent random divisions and determining their important ergodic distribution. The most important ergodic distribution is the volume (Miles, 1972). But $E(V)$ itself is not particularly informative because it only shows the scale of the model. The best representative of the nature of a random division is then the coefficient of variation, $\mathrm{cv}(V)$, which is the ratio of the standard deviation to the mean. The mean 1-projection or mean caliper diameter, $M_1$, of a domain is the mean length of its orthogonal projection onto an isotropic random line, whereas the mean 2-projection is the mean area of its orthogonal projection onto the same. For a polyhedron, $M_1(4\pi)^{-1}\sum d_e^i(\pi - \theta_i)$ where $\theta_i$ is the dihedral angles in radians. The $i$-facets of a polyhedron, $i = 0, 1, 2$, are its vertices, edges and faces respectively. The seven values of the basic integral geometric polyhedral quantities are $V$, $A$, $M_1$, $d_e$, $n_c^f$, $n_c^e$ and $n_c^v$. If a polyhedron is simple, *i.e.* each vertex is in three and only three faces, then $3n_c^v = 2n_c^e$ and from Euler's formula, $n_c^v - n_c^e + n_c^f = 2$, then it follows that the value of any one of $n_c^v$, $n_c^e$ and $n_c^f$ determines the value of the other two. The mean value of $n_c^f$ give a good idea of the overall interface structure, and so is only next in importance to the $\mathrm{cv}(V)$. The homogeneous Poisson $s$-flats in $\mathrm{R}^d$, $\mathcal{P}(s,d)$, underlies most forms of random divisions of space (*cf* Miles, 1972), in particular $\mathcal{P}(0,d)$ where $d = 2$ or 3. Under this notation the standard Poisson process is $\mathcal{P}(0,1)$.

The basic properties of $\mathcal{P}(0,3)$ are that

$$\mathrm{P}(\text{there are } n \text{ particles in } X \subset V) = e^{-\rho V}(\rho V)^n/n! \qquad (10)_{\text{ii}}$$

where $n = 0, 1, \cdots$, that is the probability is a Poisson $(\rho V)$ distribution, and that the numbers of particles in disjoint domains are mutually independent. Miles (1972) gives descriptions of various kinds of random tessellation, for example the box tessellation and particularly the generalised Johnson-Mehl model, described here as Algorithm ℵ, which includes both the standard Johnson-Mehl and the Voronoi tessellation in three dimensions as special cases.

**Algorithm 2.1** *Generalised Johnson-Mehl model, Miles (1972)*

**for** each time step **do**
  **while** more nuclei to be born **do**
    **borns** a nucleus;
    **if** the new nucleus would occupied a site already occupied **then**
      **remove** the nucleus;
    **endif**
  **endwhile**
  **for** all growing nuclei **do**
    **for** all its rays still growing **do**
      nucleus **radiates** with speed $v$;
    **endfor**
  **endfor**
  **for** all those nuclei which has just grown **do**
    **for** all its rays just grown **do**
      **if** it has met with a ray of another nucleus **then**
        **label** both rays as grown;
      **endif**
    **endfor**
    **if** all its rays are grown **then**
      **label** the nucleus as fully grown;
    **endif**
  **endfor**
**endfor** □

Similar to the Minkowski space, the stochastic nucleus birth process is in (3+1)-dimensional space, $(x, y, z; t)$-space where $t \geq 0$ which can be denoted by $\mathrm{R}^4_+$. This process is homogeneous in $(x, y, z)$ but not necessarily so in $t$ because in the actual development process it is always inhomogeneous with respect to time. Then this birth process is a Poisson point process in $\mathrm{R}^4_+$, which is inhomogeneous in $t$ and with the intensity $\alpha(t)$. For the standard Johnson-Mehl, $\alpha(t) = \alpha$ is a constant, whereas for the Voronoi $\mathcal{V}(3,3)$, $\alpha(t) = \rho\delta(t)$, where $\delta(\cdot)$ is the Dirac $\delta$ function. This model is thus a general one which can be fitted to a wide range of data by choosing some appropriate $\alpha(\cdot)$.

Each $i$-facet of the Voronoi tessellation $\mathcal{V}$, created from $\mathcal{P}(0,3)$, is an $i$-facet of $4-i$ members of $\mathcal{V}$, with each point on the $i$-facet being equidistant from the corresponding $4-i$ nuclei where $i = 0, 1, 2$. The exact values of the moments of the 3-d Voronoi tessellation in the notation used by Miles (1972) are: for first order moments, $\mathrm{E}(V) = \rho^{-1}$, $\mathrm{E}(S) = (256\pi/3)^{1/3}\Gamma(5/3)\rho^{-2/3} = 5.821\rho^{-2/3}$, $\mathrm{E}(M_1) = 4^{2/3}\pi^{5/3}\Gamma(1/3)\rho^{-1/3}/(3^{5/3}5) = 1.458\rho^{-1/3}$, $\mathrm{E}(L_1) = 12\,\mathrm{E}(M_1) = 17.50\rho^{-1/3}$, $\mathrm{E}(N_2) = (48\pi^2/35) + 2 = 15.54$, $\mathrm{E}(N_2^*) = 8$, $\mathrm{E}(N_1) = 144\pi^2/35 = 40.61$ and $\mathrm{E}(N_0) = 96\pi^2/35 = 27.07$; for a second moment, $\mathrm{E}(V^2) = 1.180\rho^{-2}$; for the aggregate of edges, $\mathrm{E}(L) = \mathrm{E}(L_1)/\,\mathrm{E}(N_1) = 0.4309\rho^{-1/3}$; for the aggregate of faces, $\mathrm{E}(A) = \mathrm{E}(S)/\,\mathrm{E}(N_2) = 0.3746\rho^{-2/3}$, $\mathrm{E}(B) = 2\,\mathrm{E}(L_1)/\,\mathrm{E}(N_2) = 2.252\rho^{-1/3}$ and $\mathrm{E}(N) = 3\,\mathrm{E}(N_0)/\,\mathrm{E}(N_2) = 5.228$; for the plane section, $\mathrm{E}_2(A) = 1/\rho\,\mathrm{E}(M_1) = 0.6859\rho^{-2/3}$, $\mathrm{E}_2(B) = (6/\pi)^{1/3}5\Gamma(2/3)\rho^{-1/3} = 3.136\rho^{-1/3}$, $\mathrm{E}(N) = 6$ and $\mathrm{E}(A^2) = 0.698\rho^{-4/3}$; and for the line section, $\mathrm{E}_1(L) = 4/\rho\,\mathrm{E}(S) = 0.6872\rho^{-1/3}$, $\mathrm{E}_1(L^2) = 0.682\rho^{-4/3}$, $\mathrm{E}_1(L^3) = 0.668\rho^{-1}$ and $\mathrm{E}_1(L^4) = 0.774\rho^{-4/3}$. In our terminology, $\rho$, $S$ and $L_1$ are respectively $\rho_c$, $A$ and $d_e$, $N_0$, $N_1$, $N_2$ and $B$ respectively $n_c^v$, $n_c^e$, $n_c^f$ and $s$. The Delaunay tessellation, on the other hand, is better understood since we know that

$$\mathrm{E}(V^k) = \frac{35\pi^{1/2}(k+2)!(2k+4)!}{256\Gamma\left(\frac{1}{2}k+2\right)\Gamma\left(\frac{1}{2}(3k+9)\right)(8\pi\rho)^k}. \qquad (11)_{\text{ii}}$$

Shape is the most fundamental geometrical property. Shape and size are all the geometrical information that remain when location and rotational effects are filtered out from an object (Dryden and Mardia, 2002), and between these two you can take the size away so that only shape remains for further analysis. When we talk about particle sizes in simulation, it is usually the case that we have already assumed some kind of particle shape. This is because the definition of size is only meaningful if you have some idea about the shape. Shape analysis works with landmarks, which are also known as anchor-, control-, design-, key-, model-, profile-, or sampling points, facets, markers, nodes, sites, fiducial markers, *etc.* Dryden and Mardia (*ibid.*) work with three types of landmarks, *viz.* anatomical-, mathematical- and pseudo landmarks. Their work could become very interesting if combined with another problem of object location (*cf* Tiyapan, 1996). A landmark can be unlabelled or labelled with a name or number. A particular member of the shape set which is used as a representative of all other members is the icon of that set. For the shape analysis in two dimensions the thin-plate spline is a convenient tool which is bijective and is analogous to the monotone cubic spline.

## § 2.1 Stereographic projection

*Stereographic projection* (*cf* Phillips, 1949) is used to represent a three-dimensional figure in two dimensions. It is useful in the study of symmetry of crystals. The stereographic projection of a cube is shown in Figure 2.1.

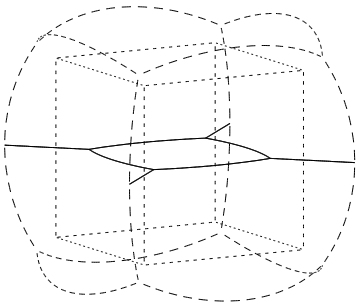


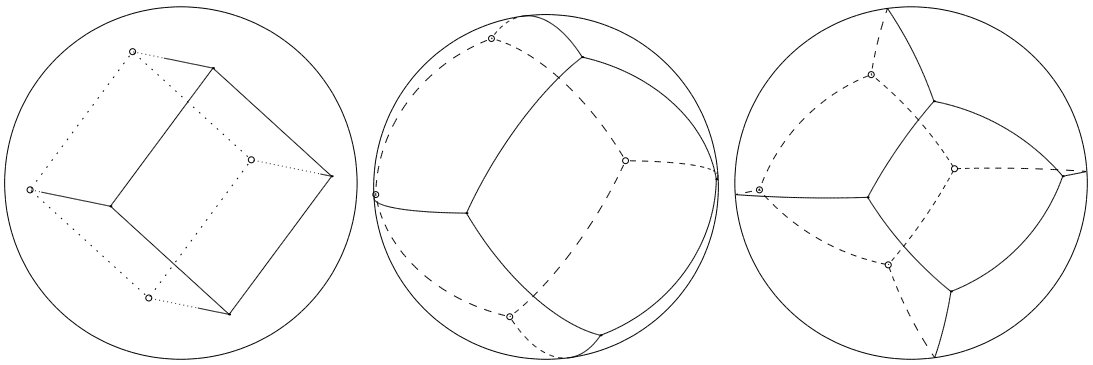
Figure *fste Stereographic projection* of a cube. A stereographic projection is drawn by first projecting each vertex on to the surface of a sphere encircling the polyhedron along the line coming from the origin of the sphere. The projected point on the sphere surface is then projected on to the plane $z = 0$ (ie. $(x, y, o)$) along the line which originates from it and goes towards the nadir point relative to the half-sphere surrounding it that rests on the $z = 0$ plane.

With this kind of projection, points that are symmetrical to each other with respect to the plane $z = 0$ are projected on to the same point on that plane. Therefore it is a normal practice to distinguish points of the two hemispheres by drawing those in one of them as dots, while drawing the rest as circles. Continuous lines going from one hemisphere to another will become discontinuous on the projected plane. In Figure 2.1, the top and the bottom squares of the cube is projected on top of each other, while the four edges parallel to the $z$-axis go to the circumference of the great circle of the projected plane first, then retrace their ways back to their vertices.

Figure 2.1 shows that, in effect, what the stereographical projection does is to bloat a polyhedron out into a spherical balloon, and then project the image obtained on the balloon on to the horizontal plane. What the second projection does is to look at the hemisphere above it with a 90° wide-angled lens from the nadir position. In order to compare the shape of the original polyhedron with the top-view of the balloon as well as the stereographical image, Figure 2.2 draws our cube rotated one radian around the vector $(0.3, 0.4, 0.866)$. Such rotation can be done by using the transformation matrix

$$M = \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & 1-2(x^2+z^2) & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & 1-2(x^2+y^2) \end{bmatrix},$$

where $[(x, y, z), w] = q$, a quaternion, $(x, y, z) = u \sin\alpha$ and $w = \cos\alpha$. Here $u = (a, b, c)$ is the direction cosine vector of the axis of rotation and $\theta = 2\alpha$ is the angle of rotation.



(a) (b) (c)

**Figure 2.2** *Rotated cube, (a) top-view, (b) top-view of its bloated sphere and (c) the stereographic projection. While (c) shows the same amount of symmetry information as the other two, it incorporates more information regarding the z-coordinate than the others, (b) being the least informative in this respect.*

The information regarding symmetries of the polyhedron is preserved by the projection. In practice, when it is applicable and possible, the sphere is positioned such that its centre coincides with the centre of symmetry of the polyhedron, the plane $z = 0$ on one of the symmetry planes, and the planes $y = 0$ and $x = 0$ on two others. Thus Figure the projection of 2.2 clearly shows the

symmetry around the centre as well as the reflection symmetries with respect to the planes $z = 0$, $y = 0$, $x = 0$, $x = y$ and $x = -y$. Figure 2.2, on the other hand, only shows the symmetry around the centre.

The stereographic projection is an example of a homeomorphism, that is to say, a mapping of one figure onto another that is both continuous and one to one.

A stereographic net is called the Wulff net. I comprises of a family of great circles at 2° intervals and a family of small circles. The great circles are equivalent to the meridians of longitude while the small circles the parallel latitudes (*cf* Cox *et al*, 1974).

The stereogram which is made up of two 2-d pictures, one for each eye, can also be used for visualising a crystal structure in three dimensions. It works by tricking or brain to see the virtual 3-d image from the input prepared for both eyes. Unlike the stereoscope which requires a viewing device, an autostereogram requires none and only a little practice. During the 1990's there has been a boom in businesses related to a certain type of stereogram referred to as SIRDS, Single Image Random Dot Stereogram, which was only a new name for the autostereogram. The boom of this business had the origin in Japan and was fueled by consumers in the east-asian countries, including Thailand. In this kind of autostereogram there are no two separate images. The two images appear as dot patterns embedded within a random pattern of dots in the background. This makes one see the 3-d images arising out of the blue amidst a seemingly chaotic random mixing of dots. SIRDS sometimes appears as another variant called Single Image Stereogram, SIS, which uses patterns in place of dots.

Stereograms and Autostereograms have no depth of field. This gives it a peculiar sharpness because all points from the highest to the deepest appear in focus at once, whereas in viewing the real world our eyes focus on one distance at a time. The reason behind the surge in the popularity of SIRDS's mentioned must be that they help relax the eyes. Because there is no special gadgets required, one needs to look at the image wide-eyed, that is to say, with one eye focusing parallel to the other, and this is very relaxing to the muscles of the eyes. By contrast, cross-eyed viewing can induce headache and eyestrain. One procedure for drawing a stereogram from a crystal or crystal model is shown in Algorithm 2.2.

**Algorithm 2.2** *A procedure for drawing stereograms from crystal models.*

**make** drawings of the crystal;

**label** all the different faces;

**select** prominent zones;

**for** every zone **do**

  **measure** all the interfacial angles;

**endfor**

**plot** the prominent zone in the primitive circle;

**mark** the centre of the stereogram;

**locate** $a$ arbitrarily;

**mark** the remaining face poles;

□

The interfacial angles are measured using a contact goniometre. The sum of the interfacial angles in a complete zone is 360°. Face poles are marks obtained from intersections of small circles with with primitive circles.

The programs on stereographic projection are collectively called `stp.m` and listed in § A.30. Some of the pictures drawn during its developmental stages are given in Figure 2.3 (a) and (b) while those for future developments in Figure 2.3 (c) and (d).

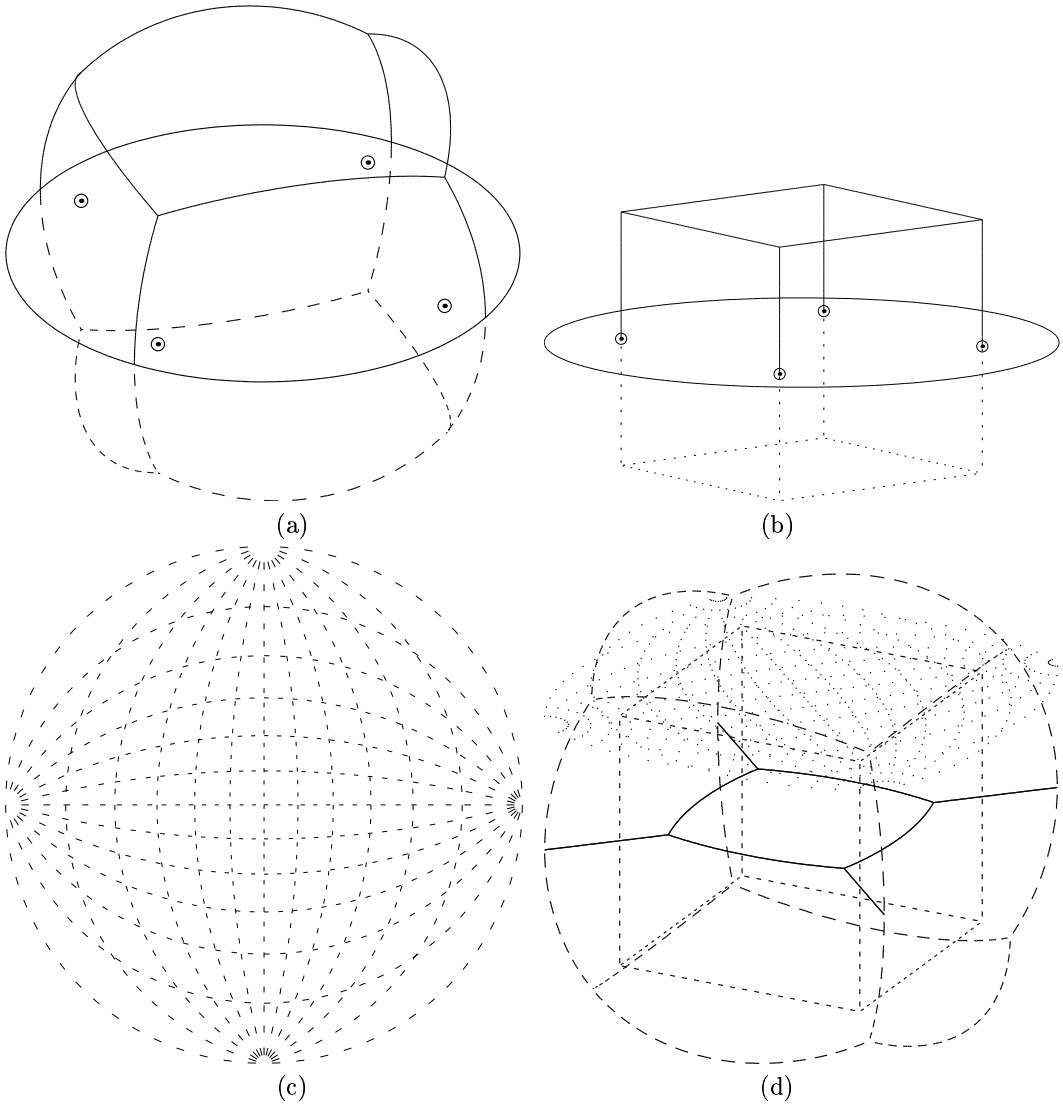



**Figure 2.3** *Developmental stages of programs for the stereographic projection;* (a) *and* (b) *for the cube,* (c) *and* (d) *the great circles.*

The parameters of any two faces of a crystal are always rationally related to each other. One way of uniquely presenting the relationship between angles of the faces and those of the crystallographic axes is by using the Miller indices. These idices are obtained by first finding intersections between a face and $a$-, $b$ and $c$ axes. Then divide by the $b$-axis intersection to get parameters of the face, and then divide these by the respective values of the parametral plane chosen. What obtained after this stage are the ratio of the parameters to those of the parametral plane, the reciprocal of which gives the Miller indices.

## § 2.2 Covering lattices

A covering lattice of any two-dimensional lattice is the lattice obtained by joining midpoints of consecutive edges together. The code in § A.7 finds covering lattices up to the eighth one and computes the total area of the cells for each case.

The square lattice is the only regular covering lattice, that is it is both the dual and the covering lattices of itself. But all polygonal tilings and tessellations can have a covering lattice, or in fact an infinite orders of covering lattices. Coverings of some lattices can be seen in § 3 (*cf* Tiyapan, 2001).

Coverings can be generalised to a general dimension $d$. In two dimensions they are lines, *i.e.* having two vertices, straight lines each of which join two lines across a corner. For three-dimensional polyhedral tessellations they are planes with three vertices, triangles each of which join three planes across a corner, in other word a coign. Then in four dimensions they may be polyhedra with four vertices, tetrahedra each of which joins four 4-d polytopes across a four-edged corner in four dimensions, and so on. In $d$ dimensions, then, perhaps they are polytopes with $d$ vertices, three-cornered $(d-1)$-polytopes each of which joins $d$ $d$-d polytopes across a $d$-edged corner in $d$-dimensions.

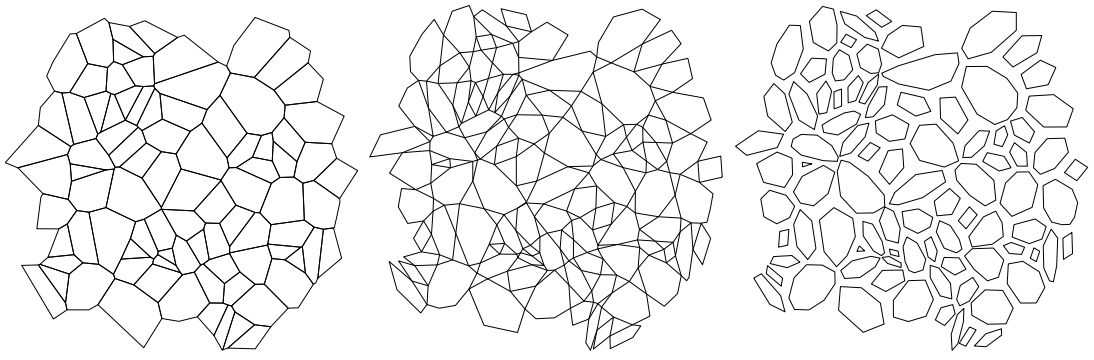
One interesting property of covering lattices is that they leave the voids intact while reducing only their size. Thus the structure and complexity of the original tessellation remain unchanged. This can be useful when we want to exclude some of the volume. In filtering membrane studies, for instance, this is ideal since all the voids still remain in the same position.

The process of covering is similar to that of shrinkage in cell in that there is a retreat away from corners. This could be because corners are hard to maintain. Surface tension is high there, and like a nook, recess, or remote, unstrategic parts of a country the cost of maintaining and governing is high. In the case of a country, conflict in such parts is analogous to the high surface tension in corners of cells.

When an empire or a metropolitan city declines, the process is similar. Such far corners are the first parts to fall into chaos. Law, order and security shrink away from them. The Roman Empire is an excellent example of this. In its hayday it reached out to every corner of Europe, however far. When it comes to the decline, it literally *pull itself together* and pull away first from those far corners, and then towards its nucleus.

Manchester is another interesting example. After the industrial revolution, and under the governance of the Conservative party, the city declined. And as it did, all the different nuclei became prominent, if only because the distance between them became more so. Thus Bolton, Altrincham and Stockport, for example, shrunk towards their nuclei, leaving behind dangerous districts where mugging, murder and crime are rampant like the Moss Side decades ago until shortly after the IRA bombing of the city centre. When an urban area fades away it does not do so suddenly but like the plant cell subject to a dewatering process or a polygonal tessellation to a covering one.

Random fluctuation can create areas of irregularity within a homogeneous and isotropic universe. These irregularities become nuclei, and from duality of the structure fine partitions start to develop around them which become Voronoi facets. Gradually but steadily the gas shrinks to form the galaxies of our present universe. Figure 2.4 (a) to (i), which are the results of the covering operator applied to a Voronoi graph eight times in succession, represent this situation.



(a) (b) (c)

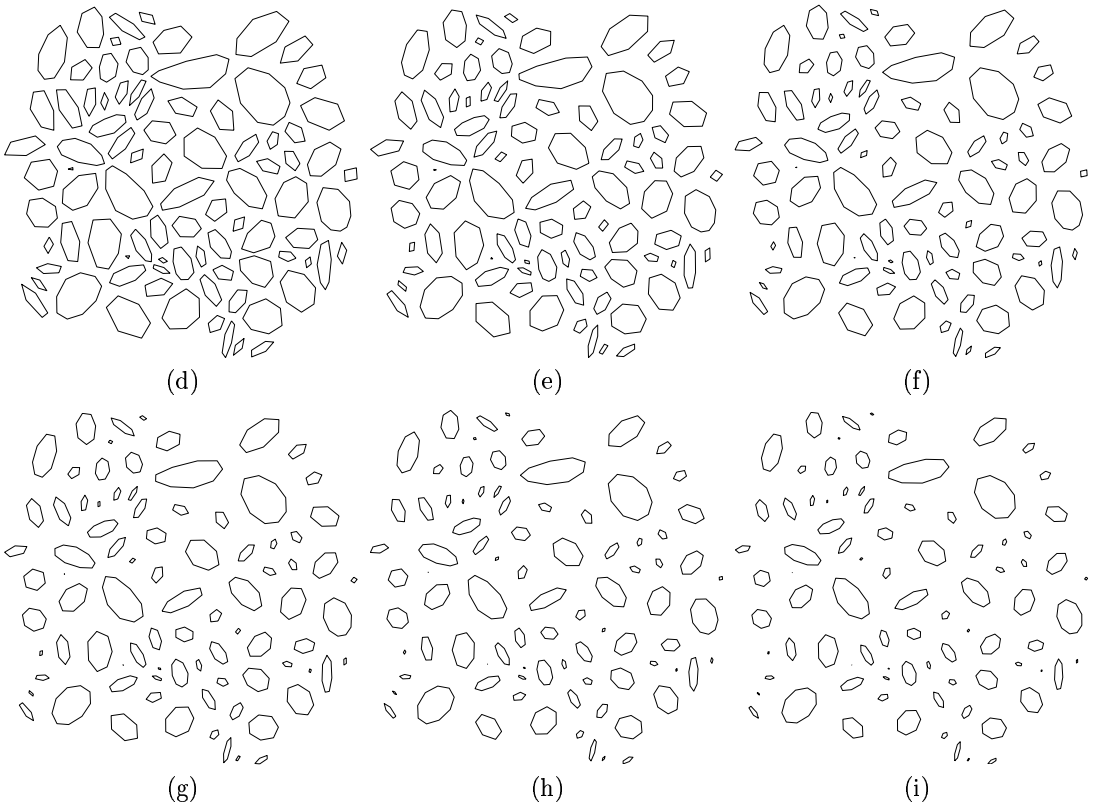



**Figure 2.4** *Covering lattice, stone pavement, or galaxies in the forming?* (a) $\mathcal{V}$, (b) $\mathrm{C}^1(\mathcal{V})$, (c) $\mathrm{C}^2(\mathcal{V})$, (d) $\mathrm{C}^3(\mathcal{V})$, (e) $\mathrm{C}^4(\mathcal{V})$, (f) $\mathrm{C}^5(\mathcal{V})$, (g) $\mathrm{C}^6(\mathcal{V})$, (h) $\mathrm{C}^7(\mathcal{V})$, (i) $\mathrm{C}^8(\mathcal{V})$.

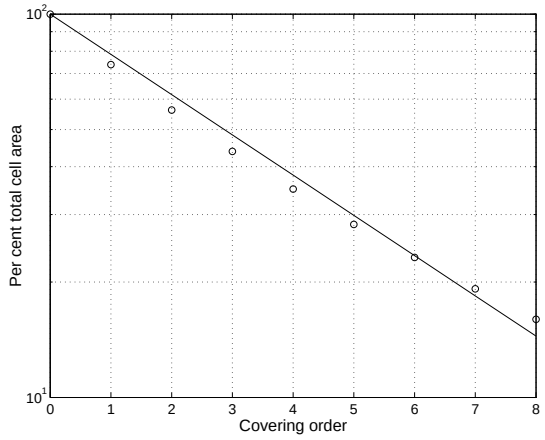



Figure 2.5 shows the area of multiply covered cells $\mathrm{C}^n(\mathcal{V})$, $n = 0$ to 8, where $\mathrm{C}^0(\mathcal{V})$ is the Voronoi graph $\mathcal{V}$. Circles are the per cent total area, and the curve is $y = 10^{-0.8x+2}$. The code `gxy.m` to find the covering contours above is given in § A.8. The area of the cells decreases from the covering operator, not linearly but with deceleration as Figure 2.5 shows.

**Figure 2.5** The area of multiply covered cells.

On taking a closer look at Figure 2.5, one can interestingly notice that eventhough all pores shrink from the application of the covering operator, some does so much quicker than others. In particular, round pores shrink but slower. The more corners a pore has the less acute are the angles, which makes it the more stable and thus able to maintain its original size.

Geometrically, one can see that the most unfortunate of all polygons is the triangle. The area of a triangles of any shape reduces by 75 per cent upon being covered. The circle is the most fortunate in this matter since it has no corners and therefore it is impossible for these to be cut. This is in accordance with our argument that corners are unstable region.

## § 2.3 Viscous fingers

Viscous fingering is a happening which occurs when a low viscosity fluid with high pressure penetrates the border of a higher viscosity fluid in a form of thin branching fingers resembling a flash of lightnings in shape. An experimental account as well as descriptive pictures and diagrams are given by Nittmann et al (1985). However, a reference made to one previous work (Hele-Shaw, 1898) was inaccurate in at least three ways, one of which is in the initials of its author, while the rest in the actual contents. The first one can easily be verified with a little research, is that the initials of the author of the paper is H. S., not J. S. S. as written there, the full name being Henry Selby Hele-Shaw (1854–1941). For the remaining two, firstly the paper by Hele-Shaw is about fluid flow pass free boundary of solid, not *viscous fingering instability.* Secondly, Hele-Shaw was an engineer who has been a professor at University college in Liverpool for 17 years, whose interest was in *layer parallel motion* (laminar) and *sinuous* (turbulent) flows pass various bodies of uniform cross section as well as in flows through channels of varying cross section. Born at Billericay, Essex, and a holder of Whitworth scholarships while a student at University college, Bristol, he was eleced to the Royal Society in 1899 because of his experiments done on streamline flow. He has successfully introduced the use of air bubbles in experiments to help portray the stream lines of the flow; his cited work mentions neither the physics of nor the application to petroleum science as the context would have us believe. However, his other and subsequent work (Hele-Shaw, 1899) does mention briefly about the importance of viscosity in nature and the difficulty of modelling the motion of viscous fluids mathematically, but nowhere in either one of these two papers did he consider the interaction between fluids of different viscosity.

The simulation for the viscous fingering problem may be related to that for the propagation through rivulets or channels that we study here in § 6.11. It may also be related to the fron tracking simulation of dendritic growth. Front tracking has been to solve various problems numerically, for example shocks, flame-, chemical reaction and solidification fronts. In this method a dynamical problem in $n$ dimensions is transformed into a set of partial differential or integro-differential equations which are then solved numerically at a finite set of points on the front or interface (Srolovitz, 1990).

Dendritic growths, on the other hand, are characterised by their rich patterns and share one property that is similar to percolation, which is that it is very difficult, if not impossible, to completely describe the mathematics of their dynamic evolution.

As in other microstructural simulations, the underlying structure where these fingerings occur can be simplified by the mean-field method, which replaces the microstructure by a typical grain, or by making a simple geometrical model to represent it, for example a Voronoi tessellation. The computer simulation can be done using Monte Carlo methods, which describes the energy of the structure in terms of the location of defects and is based on a kinetic rate, or by numerically integrating the equations of motion that describe the evolution of the defects.

## § 2.4 Crystals, quasicrystals, and polycrystals

### Crystals, quasicrystals, and polycrystals

The shape of individual grains in polycrystal and the inter granular surface can be found by stereoscopic microradiography, but using random plane section is more convenient in practice (Aboav and Langdon, 1969). The grains of polycrystals are not arranged at random but in a characteristic way which can be expressed in simple terms and seems to be scale-free (Aboav, 1970). The average number of sides of neighbours of a grain is $n_n^e = 5+8/n_e$ where $n_e$ is the number of sides of the grain. Aboav (1970) studies grains of polycrystalline magnesium oxide and finds $n_e = 5.85$. He also finds that most of the time $n_n^e > n_e$ which seems like a contradiction but is because the probability that a point lies in a grain of a particular shape depends on the size and abundance of such grains. He also finds the average number of $n$-sided grain $n_c^{n-e} \propto (n-2)$. Grains of a polycrystal are different from cells of a soap foam in that they possess a stable grain diameter, which only depends on the temperature.

This stable diameter of grain, $d_e$, is the average grain diameter at which the growth ceases, and is $d_e^{1/2} = c(T - T_0)$ (Aboav, 1971). For cadmium $T_0 = -53°\mathrm{C}$, $0°\mathrm{C} < T < 170°\mathrm{C}$. The distribution of grain size is $z = z_m \exp\left\{-\alpha^2\left[(x/x_m)^{1/2} - 1\right]^2\right\}$ where $z$ is the number of grain sections in a plane section, $\alpha$ a constant, $x$ the diameter of grain section and $x_m$ the value of $x$ at $z_m$.

There are similar patterns of grains in a polycrystalline ceramic, magnesium oxide, cadmium, *etc.* The moments of distribution above $n = 6$ is $\mu_m = \sum_n (n-6)^N f_n$ where $f_n$ is the fraction of cells with $n$ sides. The second moment $\mu_2 > 0$ unless all cells have six sides, and we have a purely topological relation $\sum_n n_n^e n_e f_n = \mu_2 + 36$. If $n_n^e \propto 1/n_e$, then $n_n^e = (6-a+b\mu^2/6)+(6a+(1-b)\mu_2)/n$. If $a = 1$ and $b = 0$, this equation is reduced to $n_n^e = 5 + (6+\mu_2)/n$ and furthermore if $\mu_2 = 0$, $n_n^e = 5 + 8n_e$. In a polycrystal the distribution of $n_e$ does not usually vary as the grains grow. Typically $\mu_2 = 2.4$. Soap foams resemble a polycrystal (Aboav, 1980), and $n_n^e = A + B/n_e$, $n_n^e = (6-a) + (6a+\mu_2)/n$, $a = 1.2$.

The growth process of both crystals and quasicrystals are nonlocal in nature, but that of polycrystals is of a multigrain growth. Crystals with a very large unit cell exists. Quasicrystals cannot grow in the same manner as playing a jig-saw puzzle (Penrose, 1989) because no matter how many steps one looks ahead there will come a point where there is a gap that none of the available basic building blocks can fit in. A *legal* tiling contains no gaps or overlaps; if it can be extended to cover the whole plane it is also a *correct* tiling. A [mistake] occurs when a tile added to a correct tiling renders it a legal but incorrect one. By using the basic units to build larger units having the same structure as these basic units and then recursively repeat the process, one can in the end reach a correct tiling. But this is also a nonlocal process since one still requires the ability to look ahead in order to make sure that the units are always structurally the same.

The fact that zeolites are crystalline rules out the randomness assumption that one may be tempted to use to model their growths as random continuum percolation. One needs to resort to the methods and tools used by geologists if one wants to study zeolites, synthetic as well as natural ones.

*Krýstallos* is the Greek word for ice. Crystallography is nowaday the science of the crystalline state which, apart from the crystals themselves, includes such seemingly unlikely materials as plastics, rubber, silk, wool, liquids and gases (*cf* Phillips, 1949). There are four kinds of symmetry axes in crystal models, corresponding to $n = 2$ to 6 in the formula for the angles $\frac{360°}{n}$ by the rotation of which a crystal will repeat itself. These axes are respectively called *diad*, *triad*, *tetrad*, and *hexad*.

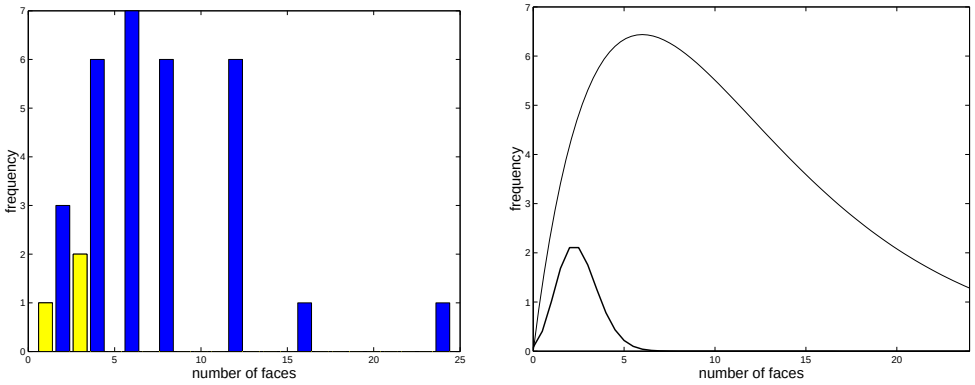
| *Crystals* | *centres of symmetry* | *planes of symmetry* | *diad* | *triad* | *tetrad* | *hexad* |
|---|---|---|---|---|---|---|
| **Cube** | 1 | 9 | 6 | 4 | 3 | |
| **Rhombohedron** | 1 | 3 | 3 | 1 | | |

**Table 2.1** *Symmetries of crystals*

The symmetries of the seven crystal systems, the Triclinic contains no axes of symmetry, Monoclinic one diad axis, Orthorhombic three diad axes, Tetragonal one tetrad ais, Cubic four triad axes, Trigonal one triad axis, and Hexagonal one hexad axis. A *monoclinic* crystal has all three axes unequal and one oblique intersection while an *anorthic*, aka *triclinic*, crystal has all three axes unequal and intersecting at oblique angles. The relative development of crystals in different forms give rise to the *habit* of crystal. Thus if one gradually truncates the coigns of a cube one will come to an octahedron, and vice versa. All the continuous transformations between the cube and the octahedron are called *cubo-octahedron*. Crystal models of different habits readily yield on inspection

the same symmetry group, but real crystals often look irregular and determination of the symmetry is based on the law of consistancy of angle and uses goniometers.

The number of faces in crystals are usually even numbers. Take for example the 33 non isometric crystal forms (Klein and Hurlbut, 1993). Here only three have their faces in odd number, *viz.* pedion which has only one face, and trigonal prism and trigonal pyramid which have three faces each. For the rest, those which have two faces are pinacoid, dome and sphenoid; four faces rhombic prism, tetragonal prism, rhombic pyramid, tetragonal pyramid, rhombic disphenoid and tetragonal disphenoid; six faces ditrigonal prism, hexagonal prism, ditrigonal pyramid, hexagonal pyramid, trigonal dipyramid, trigonal trapezohedron and rhombohedron; eight faces ditetragonal prism, ditetragonal pyramid, rhombic dipyramid, tetragonal dipyramid, tetragonal trapezohedron and tetragonal scalenohedron; twelve faces dihexagonal prism, dihexagonal pyramid, ditrigonal dipyramid, hexagonal dipyramid, hexagonal trapezohedron and hexagonal scalenohedron; sixteen faces ditetragonal dipyramid; and twenty-four faces dihexagonal dipyramid. Among these, there is none which has its faces in a number of 5, 7, 9–11, 13–15, 23, or 25 and above. In Figure 2.6 the distribution curve for even numbers of faces is a scaled Chi-square distribution $y = K \cdot (x/3)^{(\nu-2)/2} \exp(-(x/3)/2)/(2^{\nu/2}\Gamma(\nu/2))$, where the degree of freedom $\nu$ is four and $K = 35$, while the curve for odd numbers of faces is the contour of a scaled Poisson distribution $y = K \cdot \lambda^{2x} \exp(-\lambda)/(2x)!$, where $K = 12$, $\lambda = 5$, and $2x$ is any positive integer or zero.




(a) (b)

**Figure 2.6** *(a) number of faces of the thirty-three non isometric crystal forms; (b) approximation curves, the heavy line encompasses* $y = 12\exp(-5) \cdot 5^{2x}/(2x!)$ *while the lighter line* $y = (35/12)x\exp(-x/6)$.

Miller indices define the shape of crystals in terms of their faces by numbering their axial intercepts. These indices are $h = a/X$, $k = b/Y$ and $l = c/Z$ where $X$, $Y$ and $Z$ are respectively the $x$-, $y$- and $z$-intercepts. First, decide the three crystallographic axes. Then choose as the standard or parametral plane a plane that inclines against all these three axes. The shape of crystal which according to the proportion of each crystal face is called the crystal habit. It is governed by the slowest growing face and is affected by the presence of the additive solvent or impurities.. Stereoisomerism is the difference in the spatial arrangements of the same atoms and functional groups in a molecule. Enantiomers occur in pairs. They are also known as optical isomers and are mirror images of each other. Polymorphs are substances which can crystallise into different forms the chemical formulae of which are similar to one another. For example, carbon can occur as graphite, diamond, or fullerines. Isomorphous crystals always appear in one form. In pharmaceutical industries it is important to know the different properties of enantiomorphs of a drug, and to be able to grow each of them separately from the other. Crystalline materials have a tendency to cake or bind together during storage. The dimension of the particles is characterised in one of the following ways: as spheres where $V = \pi d^3/6$ and $A = \pi d^2$; as cubes, $A = 6L^2$, $V = L^3$ and $d = \sqrt{3}L$; as the maximum characteristic chord length of an irregularly shaped particle; as an equivalent diameter or characteristic length. The shape factor for the surface is $f_s = A/L^2$ while for the volume it is $f_v = V/L^3$. Their values are subjected to an inequality constraint, namely $f_s \geq \pi$ and $f_v \geq \pi/6$ where the equality is for the case of a sphere. The ratio $r_f = f_s/f_v$ called the specific surface shape factor, whereas the surface area ratio between that of the sphere and that of the particle of equal volume is called the sphericity, $\Psi$. The specific surface is $\alpha = A/V$ and onsequently $r_f = \alpha L$. The specific surface of particles is the surface area of particles per volume of particles, $\alpha_p = r_f/\bar{d}_s$, while the specific surface of bed is the surface area of particles per total volume of bed, $\alpha_b = (1-\varepsilon)\alpha_p$. The

particle size distribution is usually described by the cumulative mass fraction which is the fraction of mass of each particle size, the differential mass fraction $x = \mathrm{d}m/\mathrm{d}d$ or $\int_0^\infty x\mathrm{d}d = 1$, or the size increments. The number of particles in mass fraction $x_i$ is $n_i = x_i/(\rho_s f_v d_i^3)$, where $x_i$ is the ratio of the mass in sizes in the $i^{\text{th}}$ interval to the total mass of all particles, and $n_i$ is the product between the number, the volume, and the density of particles. The weight- or volume mean size is $\bar{d}_w = \sum_i x_i d_i \;/\; \sum_i x_i = \sum_i x_i d_i = \sum_i n_i d_i^4 \;/\; \sum_i n_i d_i^3$, where $\sum_i x_i = 1$. The mean weight- or volume size, $d_l$, is the size of each particle of a mono-disperse powder such that $f_v d_l^3 \sum_i n_i = f_v \sum_i (n_i d_i^3)$, which gives $d_w r = [\sum_i (n_i d_i^3)/\sum_i n_i](1/3) = [1/\sum_i (x_i/d_i^3)](1/3)$. The surface mean size† is $\bar{d}_s = \sum_i V_i/\sum_i S_i = \sum_i d_i S_i/\sum_i S_i = \sum_i n_i d_i^3/\sum_i n_i d_i^2 = \sum_i x_i/\sum_i (x_i/d_i) = 1/\sum_i (x_i/d_i)$. The mean surface size is the uniform size of mono-disperse particles which makes the surface area of the particles equal to the surface area of actual the powder. In other words, $\sum_i n_i f_s d_s r^2 = \sum_i n_i f_s d_i^2$, which gives $d_s r = [\sum_i n_i d_i^2/\sum_i n_i](1/2) = [\sum_i (x_i/d_i)/\sum_i (x_i/d_i^3)]$. The specific surface is the surface area of a powder per unit mass or volume, $\alpha = f_s \bar{d}_s^2/\rho_s f_v \bar{d}_s^3 = r_f/\rho_s \bar{d}_s$. The coefficient of variation (cv) describes the spread of the distribution about the mean, $\mathrm{cv} = \sigma/d$ where $d$ is the mean size and $\sigma$ standard deviation. The void fraction or voidage, $0 < \varepsilon < 1$, is the ratio of the particulate void volume to the total bed volume which comprises of the volume of voids and solids. Sieve test is used to find the distributions of particles. The mesh number of a sieve is the number of apertures per unit length of sieve, $N = 1/(L + W)$ where $L$ is the aperture size and $W$ the wire width. Crystal defects include point defects, edge- and screw dislocations.

The electrical double layer is $V_d = B' r \exp(-kh)$ where $B'$ is related to the surface charge. The interaction energy from electric field of two charged particles has two minima, $L_1 < L_2$ where $L_1$ is the primary minimum which occurs at $d_1 < d_2$.

At $L_1$ the coagulation is rapid whereas at $L_2$ it is slow. Added polymers may stabilise a colloidal system when the charged adsorbed polymer layers repel each other, or they may destabilise it by making the particles more susceptible to salts or by forming polymer bridges which flocculate the particles.

The superficial velocity, $u$, is the flowrate per cross-sectional area of bed, $u = Q/A$, and is known as the *velocity flux* or the volumetric flowrate per unit cross-sectional area. The interstitial velocity, $v$, is the true linear velocity of the fluid. It is the flowrate per unit cross-sectional area of voids, *viz.* $v = Q/(\varepsilon A) = u/\varepsilon$. An ideal particle moves in a fluid with the velocity $v$ as a function of $\mu$, $\rho$, $\rho_s$ and $d$. The momentum equation in one dimension for such particle is $\rho_s(\partial/\partial t + v\partial v/\partial z) = B + F - \partial p/\mathrm{d}z$, where $B$ is all the body forces and $F$ all the surface forces acting on the solid phase that is not included in the pressure gradient $-\partial p/\partial z$. The drag force per unit area on a single lone particle is $F/A = c_\infty \rho v^2/2$ where $F$ is force, $A$ the projected area, and $c_\infty$ the single particle drag coefficient at infinite dilution. For sphere, $F = c_\infty \pi d^2 \rho v^2/8$. The force balance equation is $W - B - F = \dot{M}$ or $mg - m'g - F = m\mathrm{d}v/\mathrm{d}t$ where $m$ and $m'$ are the mass of particle and fluid displaced, and thus the second term represents the buoyancy. Balancing the accelerative and the resistive forces leads to $(\pi/6)d^3(\rho_s - \rho)g = c_\infty \pi d^2 \rho v_t^2/8$, where $v_t$ is the terminal velocity, *i.e.* when $\mathrm{d}v/\mathrm{d}t = 0$, which gives $v_t = [4d(\rho_s - \rho)g/3\rho c_\infty]^{1/2}$. The Reynolds number of the particle is $\mathrm{Re} = \rho v d/\mu$. When $\mathrm{Re} < 0.2$ the flow is laminar and Stokes, law applies and the force balance equation becomes $(\pi/6)d^3(\rho_s - \rho)g = 3\pi\mu d v_t = F$, which gives $v_t = d^2 g(\rho_s - \rho)/18\mu$, $c_\infty = 24/\mathrm{Re}$ and the Stokes diameter $d_s t = [18\mu v_t/g(\rho_s - \rho)]^{1/2}$.

The mass flux is defined as the multiplicative product of concentration and velocity. In laminar flow, Darcy's law applies, that is $u = \mathrm{d}V/A\mathrm{d}t = K_1 \Delta P/H$, where $K_1$ is the bed permeability, a measure of the total drag force. The inverse linear relationship between the viscosity and permeability is $u = B\Delta P/\mu H$, where $B = K_1\mu$ is the permeability coefficient in $\mathrm{m}^2$. Let the pore diameter be $d_1$, its length $H_1$ and velocity $v$. Assume that the fluid in pore velocity is $v = u/\varepsilon$, the hydraulic pore diameter is $\delta$ is related to $d_1 = v_b \varepsilon/v_b s_b = \varepsilon/s_b$, and the pore length $H_1$ depends on the bed depth $H$. For laminar flow through a pipe with viscous drag the Hagen-Poisseuille equation applies, that is $v = d_1^2 \Delta P/(32\mu H_1)$. For the capillary model, $v = u/\varepsilon = \varepsilon^2 \Delta P/(K_2 s_b^2 \mu H)$. The bed specific surface area, $s_b$, and the particle specific surface area, $s_p$, are related to each other by $s_b = s_p(1-\varepsilon)$, and thus $u = v\varepsilon = \varepsilon^3 \Delta P/(K_2 s_p^2 (1-\varepsilon)^2 \mu H)$, the Carmen-Kozeny equation, from where $B = \varepsilon^3/(K_2 s_p^2 (1-\varepsilon)^2)$, where the Kozeny constant $K_2$ lies between 3.5 and 5.5 and is normally taken to be 5. In compressible beds voidage is a function of $P$ or $\Delta P$ and the Carman-Kozeny equation for $\mathrm{d}x$ is $-\mathrm{d}p/\mathrm{d}x = K_2 \mu (1-\varepsilon)^2 s_p^2 u/\varepsilon^3$. For turbulent flow $\Delta P/\Delta H = K'(1-\varepsilon)\rho u^2/(\varepsilon^3 d)$, while a linear sum of the laminar and turbulent flow is $\Delta P/\Delta H = K^3(1-\varepsilon)^2 \mu u/(\varepsilon^3 d^2) + K_4(1-\varepsilon)\rho u^2/(\varepsilon^3 d)$

† aka volume mean or Sauter mean.

where $K_3 = 150$ and $K_4 = 1.75$ (Jones, 2002). Across a filter cake the modified Darcy equation is $u = \mathrm{d}V/A\mathrm{d}t = \Delta p_c/(r\mu H)$ where $V$ is the volume of filtrate which passed through the cake. If the specific cake volume, *i.e.* the volume of cake per unit volume of filtrate, is $v_c$, then the total volume of cake is $Vv_c = HA$ where $H$ is the thickness or height of the cake. Then $\mathrm{d}V/\mathrm{d}t = A^2\Delta p_c/(r\mu v_c V)$. This, together with the Carman-Kozeny equation for packed beds $u = \varepsilon^3\Delta p_c/[5(1-\varepsilon)^2 s_p^2\mu H]$, gives the specific cake resistance $r = 5(1-\varepsilon)^2 s_p^2/\varepsilon^3$ There are two modes of filtration, the constant rate- and the constant pressure modes. In the former $\mathrm{d}V/\mathrm{d}t$ is constant and, from $V/t = A^2\Delta p_c/(r\mu v_c V)$ which is also costant, it follows that $\Delta p_c/V$ is also constant. In the latter $\Delta p_c$ is constant and equals to $\Delta p_{\max}$. It follows that $V^2 = 2A^2\Delta p_c t/(r\mu v_c) = k_1 t$ and $t/V = r\mu v_c V/(2A^2\Delta p_c = k_2 V)$ When the effect of the filter medium is considered $\Delta p$ will comprise of two components, from the cloth and from the cake. The first one includes also the particles held in the filter. The resistance is $R = r(H+L)$ where $L$ is the equivalent cake thickness of the filter. The filtration equation becomes $\mathrm{d}V/A\mathrm{d}t = \Delta p/[r\mu(H+L)]$, in other words $\mathrm{d}V/\mathrm{d}t = A^2\Delta p/[r\mu v_c(V + LA/v_c)]$ For the constant rate filtration $V_1^2 + (LA/v_c)V_1 = (A^2\Delta p/r\mu v_c)t_1$, while for the costant pressure filtration $(t-t_1)/(V-V_1) = [r\mu v_c/(2A^2\Delta p)](V-V_1) + r\mu v_c V_1/(A^2\Delta p) + r\mu L/(A\Delta p)$. Agglomeration of crystals can be on an individual- or collective basis. The first one is sometimes called primary agglomeration examples of which are parallel, dendritic and twin growths. The second one is also known as secondary agglomeration and is essentially those cases where crystals within the solution come together to form clusters. Secondary agglomeration can be perikinetic, from Brownian motion of small particles, or orthokinetic, from velocity gradients in the fluid. Crystals flocculate or coagulate loosely together, then they aggregate by starting to join one another, and then they agglomerate strongly together to become a single poly-crystalline particle.++ There are four main classes of forces, namely colloidal, stochastic, direct- and indirect systematic forces. Surface adsorps certain ionic species which in turn attract opposite ions, thus forming electrical double layers which repel each other There are three types of particle interaction, that is van de Waals, electrostatic and steric. These depend on shape and size of the particle, surface charge, solution's pH and ionic strength, temperature and the separation distance between the particles. The van der Waals forces are between molecules having closed shells and do not obey inverse square laws. Three of these are attractive, namely the orientation effect between permanent dipoles, the induction effect between a permanent- and a temporary dipoles, and the dispersion effect or London force between temporary dipoles and induced dipoles. The potential equations is $V_v = -A_{12}\mathrm{f}(h)$ where $A_{12}$ is the Hamaker constant of the material.

## § 2.5 Random lines

Miles (1964) studies the line system, $p = x\cos\theta + y\sin\theta$, where $-\infty < p < \infty$ and $0 \leq \theta < \pi$.

## § 2.6 Convex hull

The set of extreme points $E$ in some superset $S$ is the smallest subset of $S$ such that the convex hulls of both $E$ and $S$ are identical. Extreme points never lie in a triangle.

The Graham's scan algorithm (*cf* Preparata and Shamos, 1985) positions itself in the midst of the points and then scans around in one direction. It determines three points in turn, and rejects a mid point among the three if the angle made there is reflexive, *i.e.* $\alpha$ such that $\alpha \geq \pi$ in the anti-clockwise direction. In other words, an angle is a right turn if it is reflexive; it is a left turn otherwise. The algorithm is shown in Algorithm 2.3. Here $\{h\}$ is a stack which contains the points on convex hull

**Algorithm 2.3** *Graham's scan.*

$i \leftarrow 1$;
$j \leftarrow 2$;
$k \leftarrow 3$;
**while** there exist unprocessed points **do**
  **if** $\angle p_i p_j p_k \geq \pi$ **then**
    $j \leftarrow i$;
    $i \leftarrow i - 1$;
  **else**
    $i \leftarrow j$;
    $j \leftarrow k$;
    $k \leftarrow k + 1$;
  **endif**
**endwhile** □

The algorithm starts from a known extreme point. It can also start from two points which are known to be extreme, in which case the space is divide into an upper- and a lower hulls. Similarly to the Graham's scan, Jarvis's march finds one extreme point after another as it wraps a line around the convex hull. The Quickhull algorithm in two dimensions is described in Algorithm 2.4.

**Figure 2.4** *Quickhull in two dimensions.*

$l \leftarrow$ the point with the smallest abscissa;
$r \leftarrow$ the point with the largest abscissa;
$\{s\} \leftarrow$ all points above $\overline{lr}$;
$\{s\} \leftarrow$ all points below $\overline{lr}$;
**while** there remains an unprocessed $s_i$ in $\{s\}$ **do**
  $h \leftarrow$ point in $s_i$ which maximises the area of $\triangle hlr$;
  **reject** points bound by $\triangle hlr$;
  $\{s\} \leftarrow$ all points outwards from $\overline{lh}$;
  $\{s\} \leftarrow$ all points outwards from $\overline{rh}$;
**endfor** □

The divide-and-conquer algorithms divides a problem into subproblems, finds the convex hull for each one of them and then merges these together by finding the convex hull of convex hulls. Algorithm 2.5 finds the convex hull of two convex hulls.

**Algorithm 2.5** *Convex hull of convex hulls*

$p \leftarrow$ one point in $h_1$;
**if** $p$ is also in $h_2$ **then**
  $h \leftarrow$ **scan** around from $p$, and **merge** $h_1$ and $h_2$;
**else**
  $(u, v) \leftarrow$ points on $h_2$ such that $\angle upv$ is maximised;
  $c \leftarrow$ the chain from $u$ to $v$ which is furthest away from $p$;
  $h \leftarrow$ **merge** $c$ and $h_1$;
**endif** □

There are also dynamic algorithms for finding the convex hulls. In this case the input is online and one can not look ahead at the input. This kind of algorithms may be useful for online applications, for example the traffic control in real time. Two things can happen in such dynamic algorithms; points are inserted or points are deleted.

The gift-wrapping methods, which are similar to the Jarvis's march, can be extended to the general $d$ dimensions. Here a point is beneath a facet if it is on the same side of it as the hull, otherwise it is beyond it. Also, with respect to a point $p$, if $p$ is beneath all facets that contain $v$ then $v$ is concave; if $p$ is beyond the same then $v$ is reflex; otherwise $v$ is supporting.

## § 2.7 From convex hull to the Voronoi tessellation

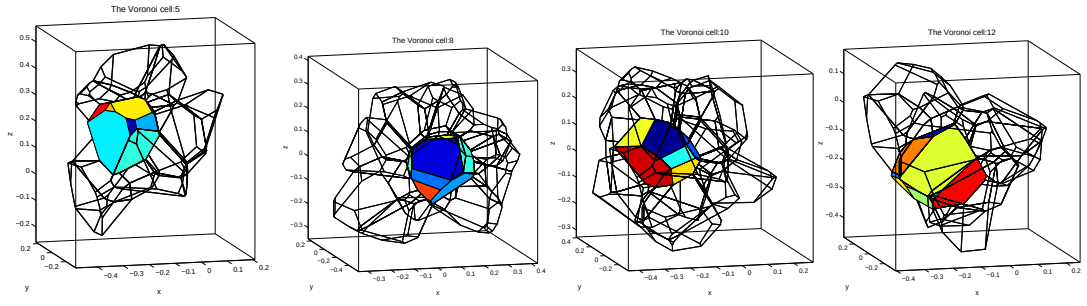
In my earlier studies of the Voronoi structure, I used the program `qhull` to create the structures, but develop my own programs on Matlab to manipulate the data and find the statistics. To test the programs, I used structures which are very small. But contrary to the literature, these very small networks turned out to have all the statistics in very close agreement with those which are much larger than themselves. This seems to have confirmed what I had experienced much earlier (Tiyapan, 1995, KNT2(ii)) that the average values of the statistics remains essentially the same for a wide range of network sizes. So a Voronoi is a Voronoi, as one could have said, almost like an Englishman.

Voronoi tessellation is the solution of a proximity problem, namely the division of the space into $n$ partitions around $n$ particles, such that all points within the $i^{\text{th}}$ partition is closest to the $i^{\text{th}}$ particle than any other particle. There are a host of proximity problems which, in the end, are related to one another and to the Voronoi problem. Some example of these are the problems concerning the nearest neighbour, the closest pair and the Euclidean minimum spanning tree. The minimum spanning tree always contains the shortest edge of the graph.

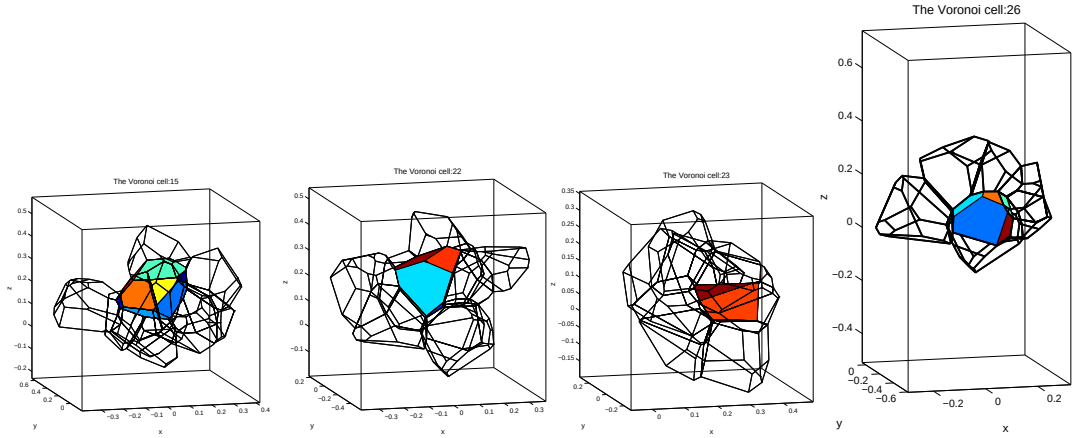
Given a convex hull containing $n$ points, one can join all the points together by straight line segments such that the whole region inside the convex hull is tessellated by the triangles formed by them. This problem is related to the nearest neighbour problem, since among all straight lines connecting to each point there is one which joins it to its nearest neighbour. Moreover, the problem is related to a problem of spatial proximity the solution of which is the Voronoi tessellation. The solutions of these two problems are dual to each other. The triangular tessellation is called the Delaunay triangulation.

Deacartes was the first person to draw a picture of a Voronoi tessellation (Descartes, 1644). In his essay, he imagine vorticities surrounding heavenly bodies. The path of an object through space, he says, passes along edges and vertices of what is now known as the Voronoi tessellation. The idea he introduced was original but the discourse philosophical, which is perhaps why his name was never associated with the tessellation which could easily have borne his name instead of that of Voronoi. Eventhough we regard Philosophy very highly as the mother of all sciences, as is probably the reason why we call our highest formal education 'Doctor of Philosophy' or Ph.D., but from our experience we could see that philosophy in our dictum generally means only one thing, that is mathematics. Therefore the tessellation is named after Voronoi because he was the first person to have written a *substantial* amount of mathematics on it. In short, Descartes has provided the idea and philosophy, Dirichlet the geometrical description and Voronoi the mathematics. Voronoi could easily have claimed having written the most amount of mathematics on the tessellation which now bears his name, than any other person to date. To see the contest between Voronoi and Dirichlet, for instance, compare their seminal papers (Dirichlet, 1850; Voronoi, 1908 (№ and №), 1909) (*cf* Tiyapan, 2001). Ironically it was Descartes' philosophy that all knowledges must be based on mathematics, so he should not have minded.

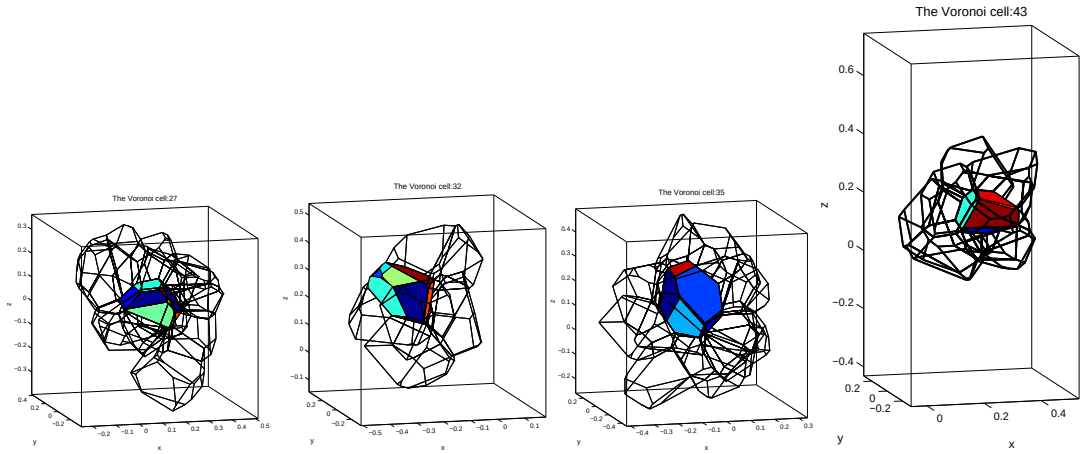
The pictures of all Voronoi cells in a group of 71 cells are given in Figure 2.7. These are all the inner cells of a larger group of 200, the rest of which are cells on the boundary.



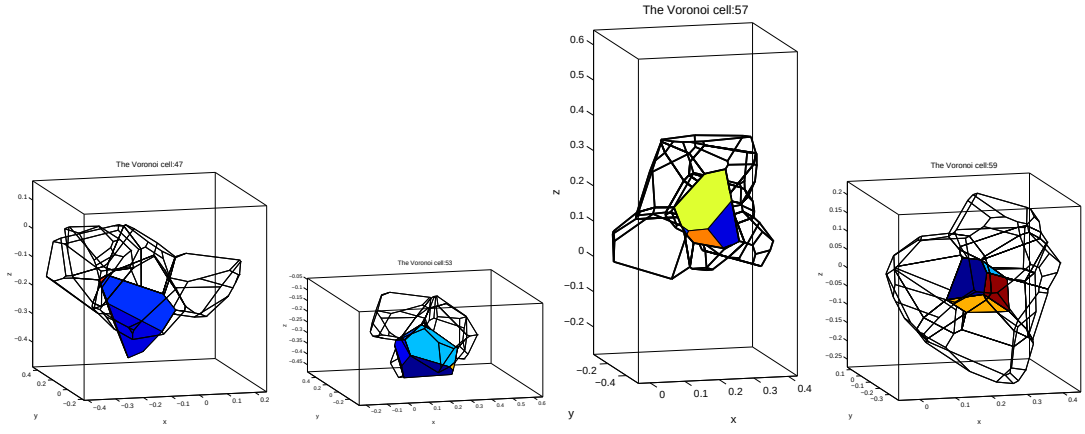

(a) (b) (c) (d)



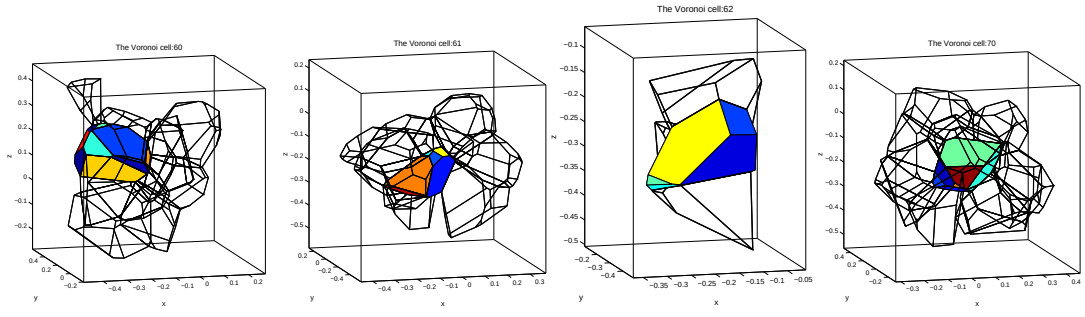

(e) (f) (g) (h)



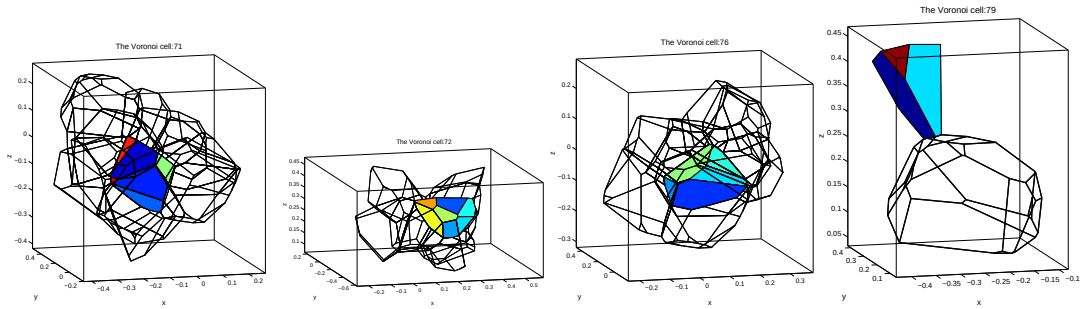

(i) (j) (k) (l)



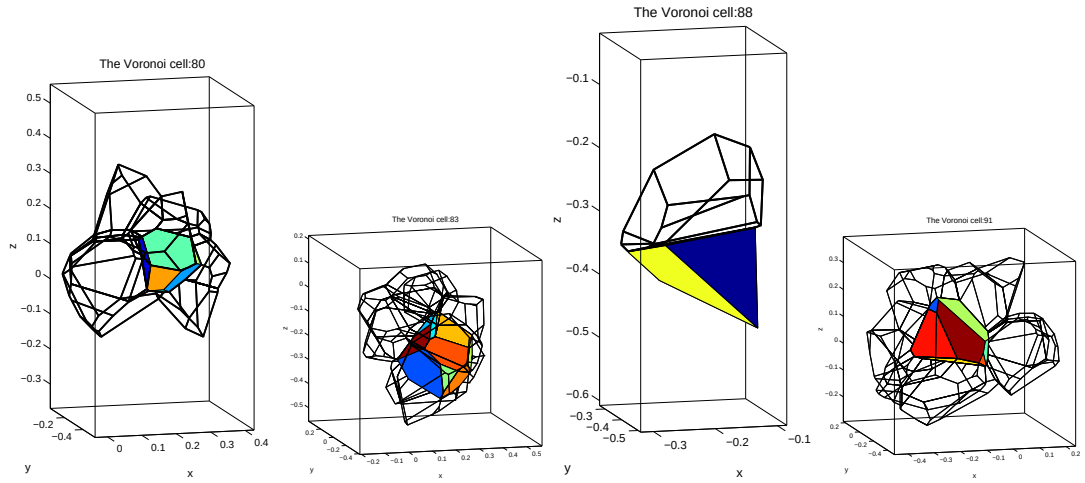

(m) (n) (o) (p)



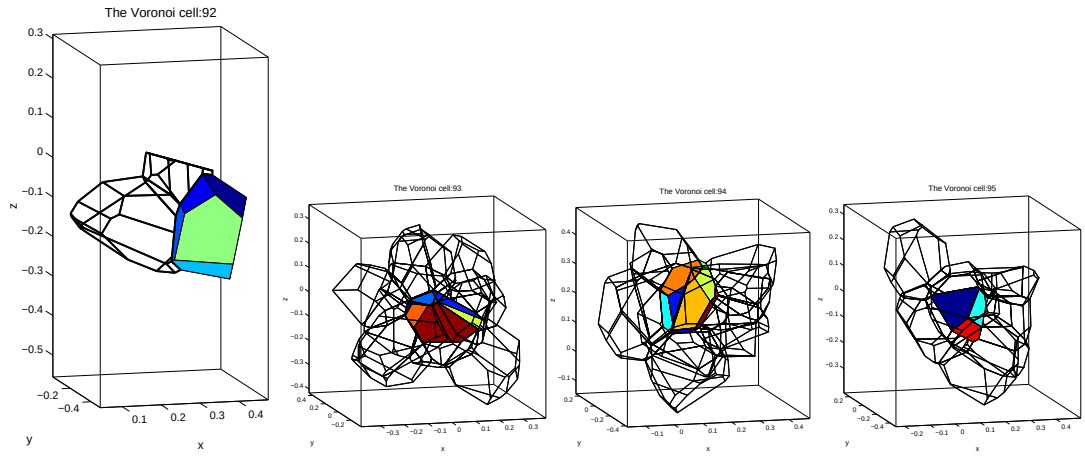

(q) (r) (s) (t)



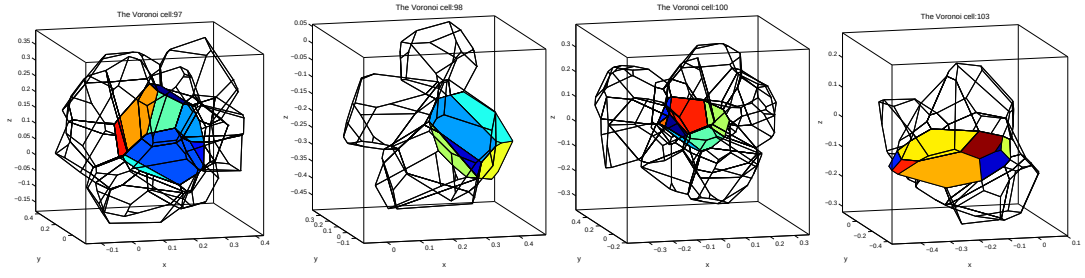

(u) (v) (w) (x)



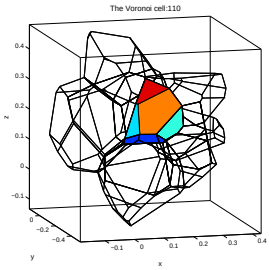

(y) (z) (A) (B)



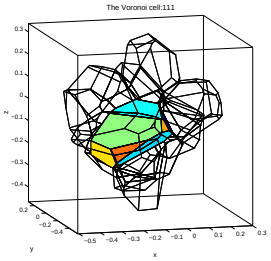

(C) (D) (E) (F)



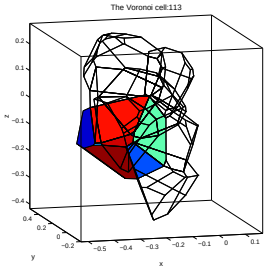

(G) (H) (I) (J)



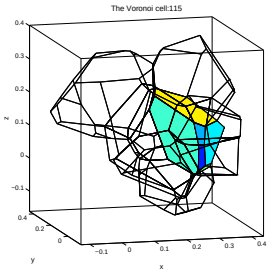

(K)



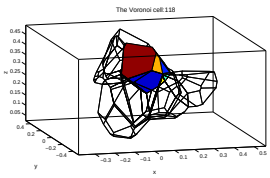

(L)



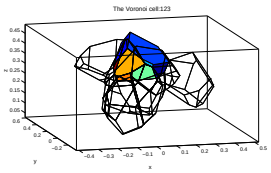

(M)



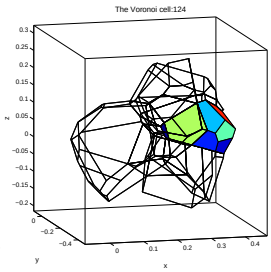

(N)



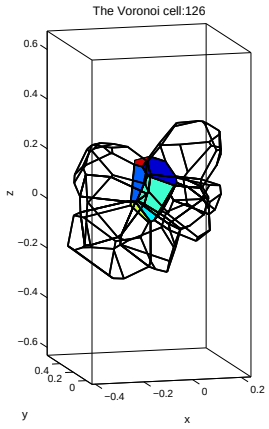

(O)



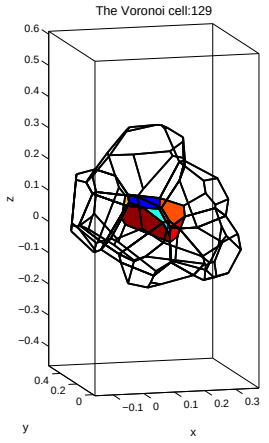

(P)



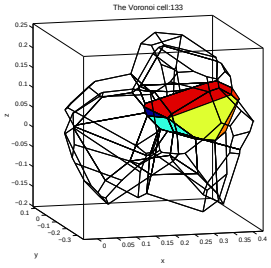

(Q)



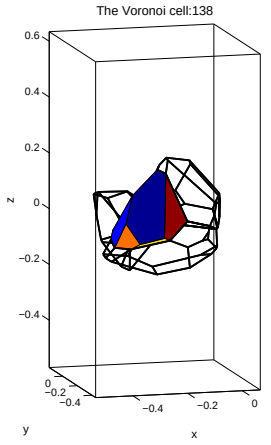

(R)



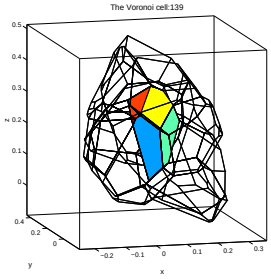

(S)



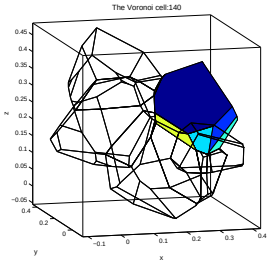

(T)



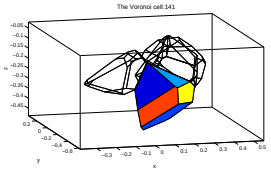

(U)



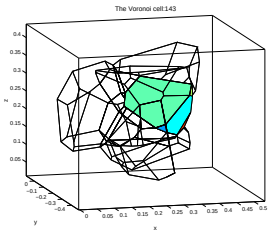

(V)



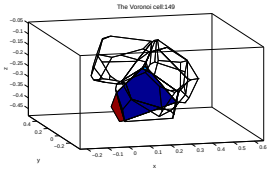

(W)



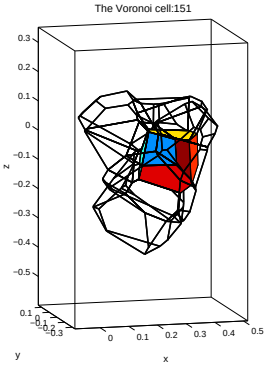

(X)



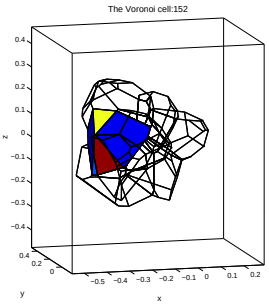

(Y)



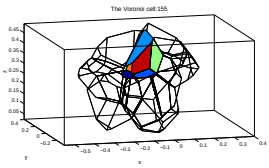

(Z)



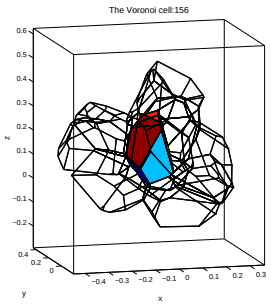

(aa)



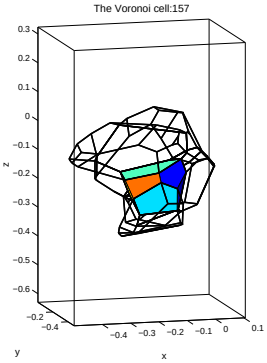

(ab)



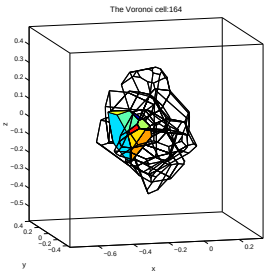

(ac)



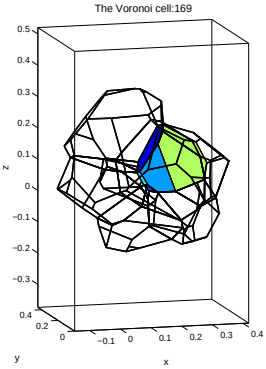

(ad)



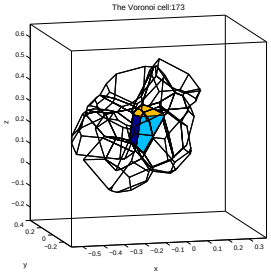

(ae)



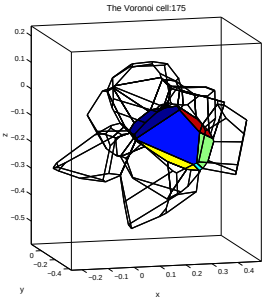

(af)



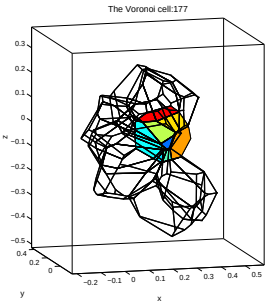

(ag)



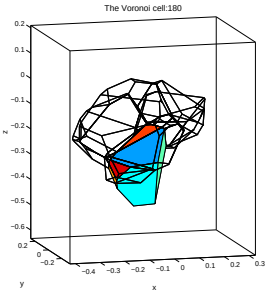

(ah)



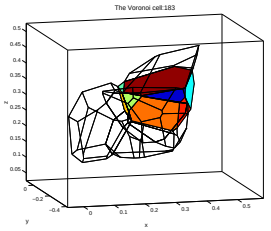

(ai)



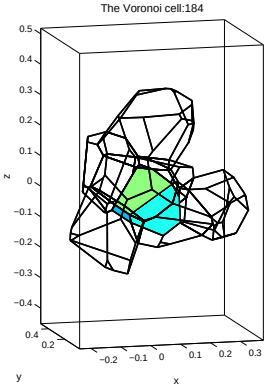

(aj)



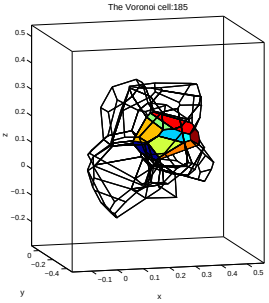

(ak)



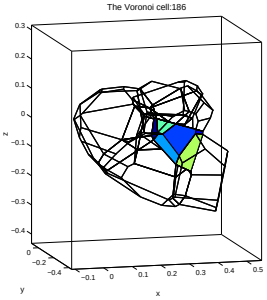

(al)



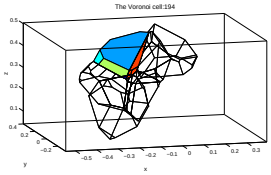

(am)



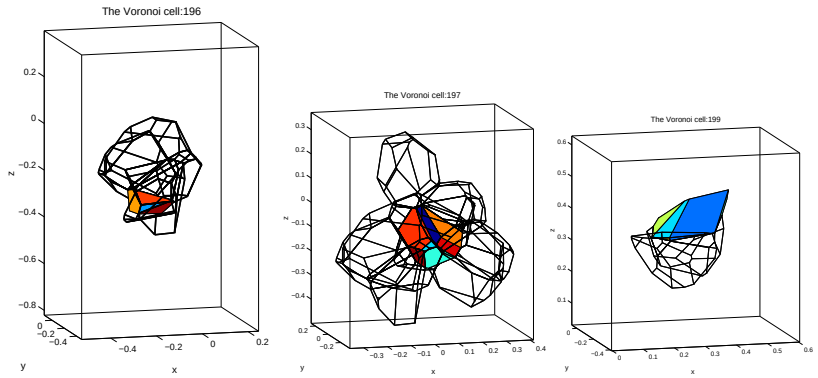

(an)




(ao)




(ap)




(ar) (as) (at)

**Figure 2.7** *The 71 Voronoi cells in an aggregation.*

The statistics of these 71 cells are the following. There are originally 200 cells, but these statistics represent 71 inner cells among these. The rest, 129 cells, are on the boundary and are not considered. The abbreviation 'mn' stands for the mean-, while 'cb' the cube normal.

**Number of faces per inbound cell**
Mean: 14.2113
Variance: 11.0262
Minimum: 6
Maximum: 23
Second moment: 10.8709
Third moment: 14.0161
Kurtosis: 3.109624
Geometric mean: 13.8246
Harmonic mean: 13.4227
Median: 13
Mean absolute deviation: 2.7169

**Number of vertices per inbound cell**
Mean: 24.4225
Variance: 44.1046
Minimum: 8
Maximum: 42
Second moment: 43.4834
Third moment: 112.1287
Kurtosis: 3.109624
Geometric mean: 23.5039
Harmonic mean: 22.5058
Median: 22
Mean absolute deviation: 5.4338

**Area of surface of inner cell**
Mean: 0.1599
Variance: 0.0024
Minimum: 0.0628
Maximum: 0.2660
Second moment: 0.0023
Third moment: $7 \times 10^{-6}$
Kurtosis: 2.3015
Geometric mean: 0.1520
Harmonic mean: 0.1435
Median: 0.1602
Mean absolute deviation: 0.0398

**Area of surface of inner cell (mn)**
Mean: 1.000000
Variance: 0.0926
Minimum: 0.3928
Maximum: 1.6634
Second moment: 0.0913
Third moment: 0.0017
Kurtosis: 2.3015
Geometric mean: 0.9505
Harmonic mean: 0.8972
Median: 1.0018
Mean absolute deviation: 0.2489

**Area of surface of inner cell (cb)**
Mean: 0.3031
Variance: 0.0085
Minimum: 0.1191
Maximum: 0.5042
Second moment: 0.0084
Third moment: $4.8 \times 10^{-5}$
Kurtosis: 2.3015
Geometric mean: 0.2882
Harmonic mean: 0.2720
Median: 0.3037
Mean absolute deviation: 0.0754

**Volume of inbound cell**
Mean: 0.0261
Variance: 0.0018
Minimum: $5.5944 \times 10^{-3}$
Maximum: 0.3315
Second moment: 0.0017
Third moment: 0.0004
Kurtosis: 42.2383
Geometric mean: 0.0183
Harmonic mean: 0.0153
Median: 0.0167
Mean absolute deviation: 0.0177

**Volume of inbound cell (mn)**
Mean: 1.0000
Variance: 2.5764
Minimum: 0.2146
Maximum: 12.7188
Second moment: 2.5402

Third moment: 24.1689
Kurtosis: 42.2383
Geometric mean: 0.7012
Harmonic mean: 0.5879
Median: 0.6408
Mean absolute deviation: 0.6795

**Volume of inbound cell (cb)**
Mean: 1.0000
Variance: 2.5764
Minimum: 0.2146
Maximum: 12.719
Second moment: 2.5402
Third moment: 24.1689
Kurtosis: 42.2383
Geometric mean: 0.7012
Harmonic mean: 0.5879
Median: 0.6408
Mean absolute deviation: 0.6795

**Number of vertices per inbound face**
Mean: 5.1114
Variance: 2.1650
Minimum: 3
Maximum: 10
Second moment: 2.1619
Third moment: 1.8902
Kurtosis: 3.0234
Geometric mean: 4.9065
Harmonic mean: 4.7086
Median: 5
Mean absolute deviation: 1.1655

**Perimeter of inner face**
Mean: 0.3899
Variance: 0.0434
Minimum: 0.0017
Maximum: 1.0067
Second moment: 0.0433
Third moment: $9.88 \times 10^{-4}$
Kurtosis: 2.3090
Geometric mean: 0.3071
Harmonic mean: 0.1497
Median: 0.3940
Mean absolute deviation: 0.1720

**Perimeter of inner face (mn)**
Mean: 1.0000
Variance: 0.2854
Minimum: 0.0043
Maximum: 2.5819
Second moment: 0.2850
Third moment: 0.0167
Kurtosis: 2.3090
Geometric mean: 0.7877
Harmonic mean: 0.3839
Median: 1.010419
Mean absolute deviation: 0.4410

**Perimeter of inner face (cb)**
Mean: 0.3288
Variance: 0.0308
Minimum: 0.0014
Maximum: 0.8488
Second moment: 0.0308
Third moment: 0.0006
Kurtosis: 2.3090
Geometric mean: 0.2590
Harmonic mean: 0.1262
Median: 0.3322
Mean absolute deviation: 0.1450

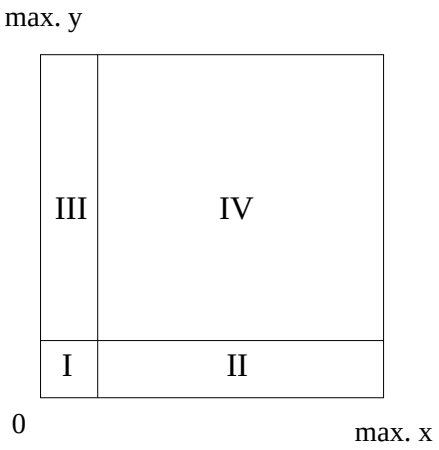
**Area of faces of inner cell**
Mean: 0.0110
Variance: 0.0001
Minimum: $4.0594 \times 10^{-8}$
Maximum: $7.2892 \times 10^{-2}$
Second moment: $1.22 \times 10^{-4}$
Third moment: $2 \times 10^{-6}$
Kurtosis: 5.2926
Geometric mean: 0.0045
Harmonic mean: $2.2 \times 10^{-5}$
Median: 0.0074
Mean absolute deviation: 0.0087

**Area of faces of inner cell (mn)**
Mean: 1.0000
Variance: 1.0085
Minimum: $3.6840 \times 10^{-6}$
Maximum: 6.6151
Second moment: 1.0071
Third moment: 1.4244
Kurtosis: 5.2926
Geometric mean: 0.4086
Harmonic mean: 0.0020
Median: 0.6715
Mean absolute deviation: 0.7933

**Area of faces of inner cell (cb)**
Mean: 0.1253
Variance: 0.0158
Minimum: $4.6179 \times 10^{-7}$
Maximum: 0.8292
Second moment: 0.0158
Third moment: 0.0028
Kurtosis: 5.2926
Geometric mean: 0.0512
Harmonic mean: 0.0003
Median: 0.0842
Mean absolute deviation: 0.0994

## § 2.8 Tilings and patterns

All two dimensional lattices are tilings. The *kagome* latticedeserves some mention here since it is a first-order covering lattice of the honeycomb lattice, which in turn is dual to the triangular lattice. Some physicists conjecture that the name originates from a name of a person. Apart from a few exceptions from the reason that the word is written in the wrong script, those scientists who are japanese should know that the word from their own language which means, according to the dictionary by Nelson (1962), *basket interstices* or *woven bamboo pattern*. My own version of the translation is simply *basket pattern* (*cf* Tiyapan, 2001), *kago* (*rad take* and *tatsu*) meaning *basket* and *me* a radical word for *eye* which also means *pattern*. Many researchers and students, blindly following an established practice, often spell it as *kagomé* (*cf*, for example, Tiyapan, 1995) with an acute accent or *l'accent aigue* like a french word.




For the case of two dimensions, Algorithm 2.6 gives the algorithm for producing patterns in general. It is used for producing the 2-homohedral tilings in § 4.9 and is the basis of the program in § A.6. In this algorithm pseudo-prototiles are produced which fill the space. We shall call these our unit tiles. There are four types of these unit tiles corresponding to the four sections or groups as shown in Figure 2.8. The first group contains only one tile, *i.e.* the one at the origin, the second one contains those unit tiles to be put at the bottom row, the third one yet those to be put at the left-most column. The rest and majority of tiles belong to the fourth group.

**Figure 2.8** Four groups of tiles.

Vertices in each unit tile are divided into five groups, namely one for the boundary in each direction, *i.e.* north, east, west and south, and the fifth those in the midst of the unit tile. This is essentially in order to avoid creating a vertex twice, which would have resulted in duplicates. Those unit tiles in the fourth group take their left-most vertices from the unit tile to their left and their bottom-most vertices from the unit tile immediately below it. Unit tiles in the third group create their own left-most vertices but still take from the unit tile below them their bottom-most ones. On the other hand, unit tiles in the second group create their bottom-most vertices while taking their left-most ones from their neighbour who lives on their left. Lastly the only unit tile of the first group creates all its vertices, which include the bottom-most and left-most ones.

There are three types of edges, namely internal-, left and bottom edges, the latter two of which link respectively to vertices in their left and bottom neighbours. And then, there are four types of cells, namely middle, south, west and south-west cells, the latter three of which contains the other half of its edges in their neighbour in the corresponding direction. Each unit tile is divided into grids by lines parallel to $x$ and $y$ axes. All vertices lie on some intersection of these grids.

**Algorithm 2.6** *Tilings in two dimensions.*

**for** every unit tile **do**
  **decide** its type;
  **assign** all vertices according to its type;
  **connect** bonds according to the connection rules;
  **define** cells by their vertices;
**endfor** □

When faced with an unfamiliar tiling, we first decide upon their pseudo-prototile, that is their unit tile. Then list the coordinates of their vertices. And then list the bonds, that is the numbers of

the two vertices that define each of them. We also have to divide the bonds into groups as mentioned, and include this information in our input.

The program keeps the two lists of coordinates, one for each axis. Coordinates of vertices are then referred to in grid numbers instead of the actual lengths concerned. This helps reading and using the program and procedure simpler. It marks the division between delight and despair. Also, the group information can then be represented in the program as a mapping of border vertices from a previous unit to the present one.

A cross-border bond normally has one end on the border. However there are those neither ends of which are on the border. These bonds make up a separate group of their own, or rather they make up two groups corresponding to the two groups, II and III.

I tried this program on a few well known regular lattices, with reasonably satisfactory results. Because these lattices are small, their coordination numbers $x$ differ from the true values. This must be due to the frayed rims of the networks we created. Therefore the discrepancy is systematic and there is nothing to worry about, or rather there is nothing one can do about it unless one makes changes to the program. Moreover, the problem will lessen but will not go away when our network size becomes larger. As the four sides that make up the border are in total $4d/d^2$, *i.e.* $4/d$ parts of the area, increasing the network size by one hundred times, *i.e.* $d = 10$, probably would reduce the error by 60 per cent.

It is possible to improve the program as regarding to this problem, and this is the plan for the future work. At present, this shortcoming will have some effect on the values of the six percolation probabilities obtained from each system simulated.

For the first test, the program was run on a square lattice. The six representative networks produced, which subsequently become the ground for the corresponding six values of percolation thresholds, have their statistics as follows: $n_C = 100$, $x_C = 6.8400$, $n_B = 342$, $x_B = 12.4211$, $n_c = 100$, $x_c = 3.6000$, $n_b = 180$, $x_b = 5.3778$, $n_v = 121$, $x_v = 3.3058$, $n_e = 200$ and $x_e = 5.4100$. Then from $2 \times 5$ runs, *i.e.* two runs for each of the five permuted list of blockages, the values of percolation thresholds are $p_C = 0.3920 \pm 0.0627$, $p_B = 0.2459 \pm 0.0373$, $p_c = 0.5610 \pm 0.1272$, $p_b = 0.4806 \pm 0.0750$, $p_v = 0.5942 \pm 0.0802$ and $p_e = 0.5040 \pm 0.0740$. To explain the results, the capital $C$ and $B$ in the subscript mean respectively cells and bonds when neighbours mean that they share at least one vertex. When in lower case letters $c$ and $b$ mean respectively cells and bonds when neighbourhood means sharing at least one edge. Neither the vertices nor the edges, respectively $v$ and $e$, is ambiguous since the former has zero dimension while the latter has only one. In this case, as it is in general, $p_C$ and $p_B$ are nothing that one normally talks about, while $p_c$ and $p_b$ should be the same as $p_v$ and $p_e$ in that order. Because all connections end at the boundary, it is to be expected that the values of all $x$'s are lower than their exact values for an infinite network. The results above show that, for this case at least, $x_c$ falls short of its exact value by 10 per cent, while similarly $x_b$ by 10.37, $x_v$ by 17.35 and $x_e$ by 9.83 per cent. Because $c$ and $b$ naturally form one pair while $v$ and $e$ another, it is interesting to note that $x_c$ should fare better than $x_v$ while on the other hand $x_e$ is more accurate than $x_b$. For the probability values, $p_c$ is off the mark by (*cf* § 4) $-5.36$ per cent and $p_b$ by $-3.9$ per cent, while $p_v$ does so by 2.5 per cent and $p_e$ by a mere 0.8 per cent. The first one of these pairs seems to be on the lower side while the second one, on the other hand, is on the higher end and more accurate than the first. This first test gives a result in accordance with our expectation that for vertices and edges the results should be more accurate than those from cells and bonds since the first two come from the input, while the last two are secondary values derived up from them by the program. Figure 2.9 shows the networks simulated for this test.

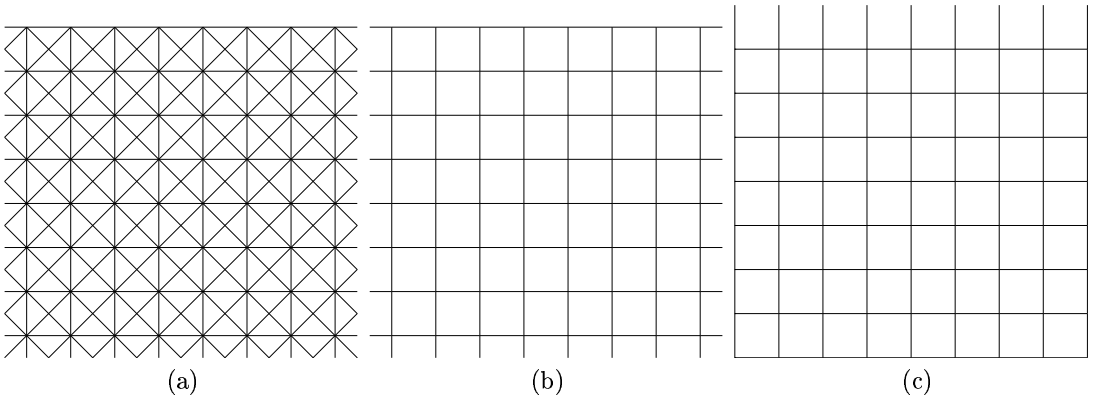



**Figure 2.9** *Percolation of networks related to the square lattice. Networks comprising of* (a) *Cells and Bonds,* (b) *cells and bonds, and* (c) *vertices and edges.*

The next test is on the honeycomb lattice. Here the results obtained from the simulation is $n_C = n_c$ 202, $x_C = x_c$ 5.4554, $n_B = n_b$ 551, $x_B = x_b$ 9.2777, $n_v = 479$, $x_v = 2.8058$, $n_e = 672$ and $x_e = 3.8244$. For the probabilities of percolation $p_C = p_c$ $0.4889 \pm 0.0704$, $p_B = p_b$ $0.3342 \pm 0.0385$, $p_v = 0.6833 \pm 0.0369$ and $p_e = 0.6382 \pm 0.0410$. These results are obtained from $2 \times 10$ simulations in the case of statistics on cells and bonds, whereas in the case of vertices and edges they are obtained from $2 \times 5$ runs. Their pictures are shown in Figure 2.10.

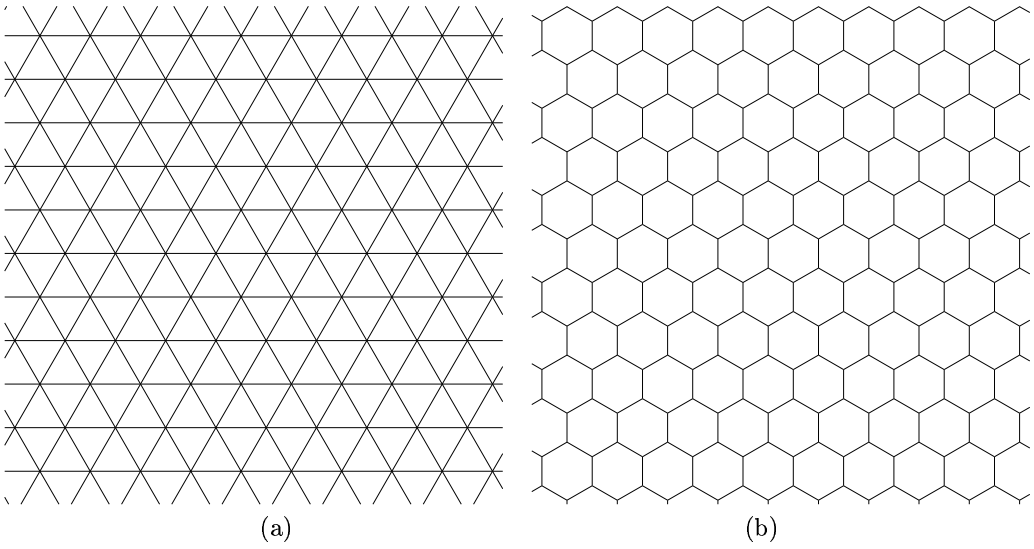



**Figure 2.10** *Percolation of networks related to the honeycomb lattice. Networks are made up of* (a) *Cells and Bonds, or cells and bonds,* (b) *vertices and edges.*

For the honeycomb lattice, the errors are for $p_c$ $-2.22$, for $p_b$ $-3.77$, for $p_v$ $-1.85$ and for $p_e$ $-2.22$ per cent. The errors for the network statistics are for $x_c = -9.08$, $x_b = -7.22$, $x_v = -6.47$ and $x_e = -4.39$ per cent.

The Kagome lattice is the next test. Here the statistics obtained for the networks are $n_C = 230$, $x_C = 6.4087$, $n_B = 737$, $x_B = 13.1452$, $n_c = 230$, $x_c = 3.2783$, $n_b = 377$, $x_b = 5.2944$, $n_v = 316$, $x_v = 3.7342$, $n_e = 590$ and $x_e = 5.7017$. The percolation probabilities are $p_C = 0.4222 \pm 0.0734$, $p_B = 0.2463 \pm 0.0482$, $p_c = 0.6548 \pm 0.0787$, $p_b = 0.5332 \pm 0.0532$, $p_v = 0.6760 \pm 0.0288$ and $p_e = 0.5309 \pm 0.0651$. Here comparison with literature is already limitted as published data begins to be rare. We can compare with exact values and say that $x_v$ obtained contains an error of $-6.65$ per cent, $p_v$ of 3.68 per cent and $p_e$ of 1.70 per cent. Figure 2.11 shows the networks created and simulated.

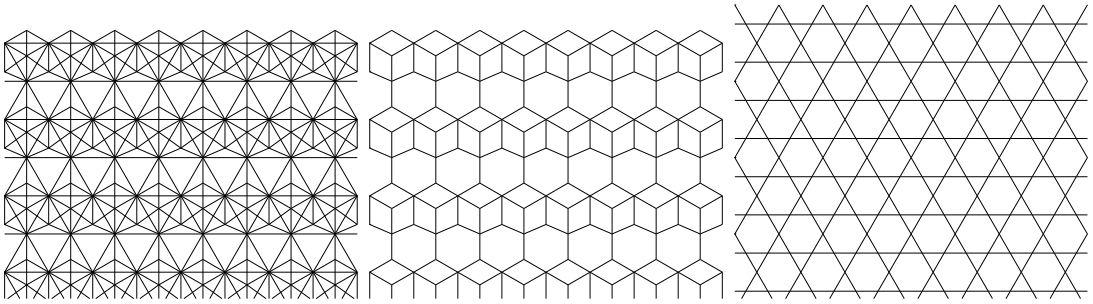


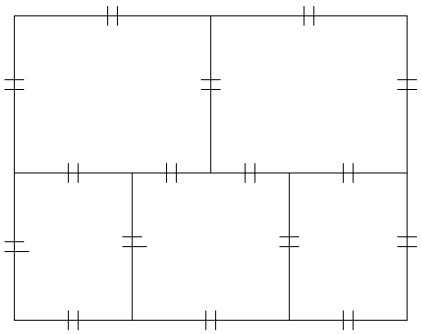

**Figure 2.11** *Percolation of networks related to the basket pattern, kagome lattice. The networks of* (a) *Cells and Bonds,* (b) *cells and bonds, and* (c) *vertices and edges.*

From these tests we can see that, even for a rather small size, our programs for creating the network and finding percolation show aproximately five per cent error.

# 3

## § 3. Voronoi tessellation

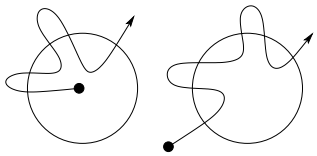
In a puzzle of Figure 3.1 there are five rooms with doors in the position as shown. The problem is whether one can walk through every door only once and the answer according to the graph theory is *no*, because there are more than two rooms which has an odd number of doors. The proof of Theorem 3.1 was from Komsan Bajáraṿáṇijẏ around 1989. From this, when one plays a puzzle like that of Figure 3.1 one always starts off from a room with an odd number of doors and ends in another such room. This is the same thing as saying that one starts and ends outside rooms with an even number of doors. Therefore the number of the latter is of no consequence, but that of the former is crucial for the existence of a solution and must never be any number other than two.



Puzzle of five rooms with doors. In Figure 3.1 there are three rooms with five doors, two with four, and one with nine. There are here four rooms with an odd number of doors. Starting off from one of these four one can only end up in one of the other three, which leaves the remaining two rooms unaccounted for. In other words at least two doors will necessarily remain unvisited.

**Figure 3.1** Rooms with doors puzzle.

**Theorem 3.1.** *A travel along a network can only starts and ends at nodes which have an odd coordination number.*



**Proof.:** Looking at Figure 3.2 if one starts from inside a room with an odd number of doors, one always ends up outside it. On the other hand one always ends up inside a room with an even number of doors if one starts of from it. In the second picture such a room is all the area outside the circle.

**Figure 3.2** Departure and arival rooms..

□

The following Corollaries 3.1, 3.2 and 3.3 assume nondegeneracy of the Voronoi network. Such a path as mentioned in these corollaries is also called *self-avoiding.*

**Corollary 3.1[1].** *There can be no path which traverses all edges of a Voronoi graph only once.*

**Proof.:** This follows from Theorem 3.1 because a two dimensional Voronoi network has a coordination number three. □

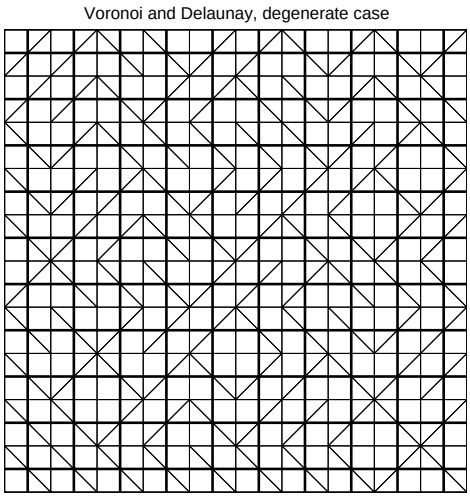
**Corollary 3.1[2].** *On a three dimensional Voronoi structure there always exists a path that runs through every edge once and only once.*

**Proof.:** This also readily follows from Theorem 3.1 since a three dimensional Voronoi network has a coordination number of four. □

**Corollary 3.1[3].** *Take any Voronoi cell of the three dimensional network, it is impossible to walk through all its edges without repeating some of them.*

**Proof.:** Again from Theorem 3.1 and because the surface of a Voronoi polyhedron is a two dimensional network of polygons which has the coordination number three. □

Jerauld *et al* (1984) compared the Voronoi, with the triangular networks and found that the bond percolation probability of the former is 4.3% or 0.015 less than that of the latter, small site clusters more, and small bond clusters less likely.




When a square lattice was fed to *voronoi* and *delaunay* in Matlab, by the program `degen.m` in § A.13, there were error messages saying that points were collinear and possibly triangulation is incorrect. This case, Figure 3.3, is degenerative.

**Figure 3.3** Voronoi from degenerative data..

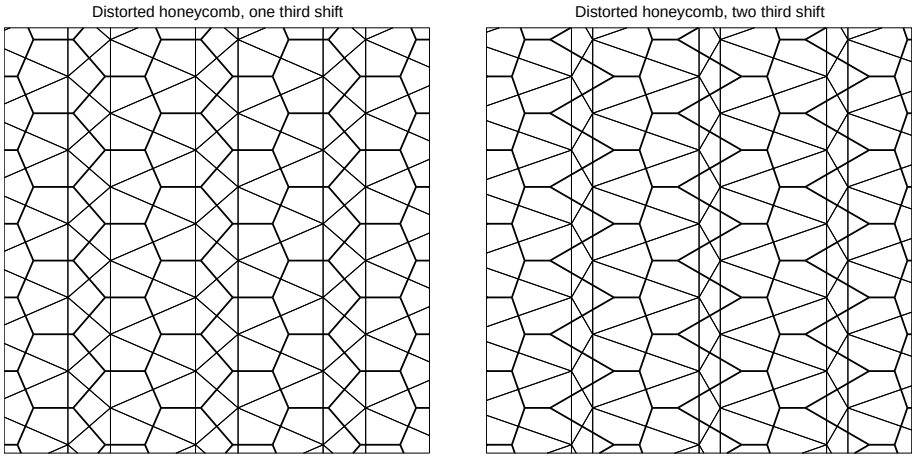



**Figure 3.4** *The honeycomb or hexagonal lattice whose alternate y-plane has been shifted (a) one-third, and (b) two-third respectively. Triangulation is shown with thinner lines. The program used is* `honey.m`

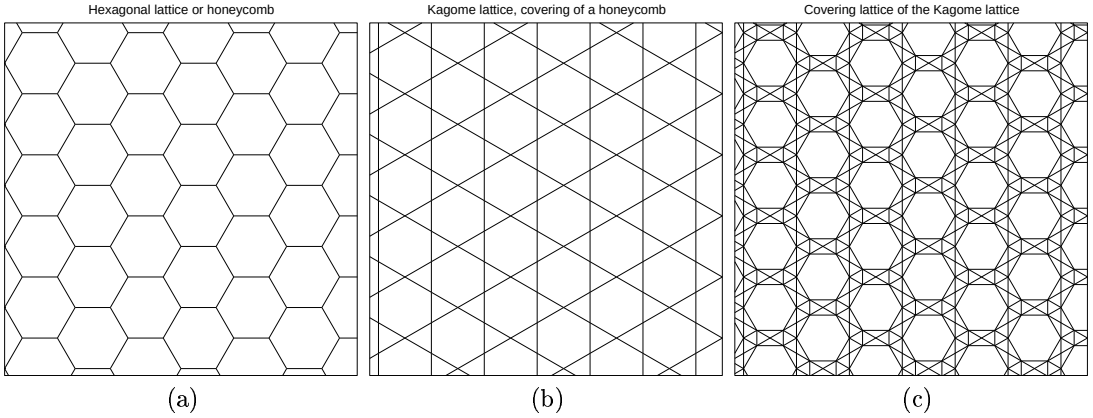



**Figure 3.5** (a) *The hexagonal lattice,* (b) *its covering lattice (Kagome), and* (c) *the covering lattice of its covering lattice (i.e. covering of Kagome).* (`cover.m`, `covers.m`, *and* `coverss.m`)

If we indicate by $C_v^n(x)$ the $n^{\text{th}}$-order covering lattice of a lattice $x$, then the first picture is Hexagonal, the second one Kagome or $C_v^1$(Hexagonal) and the third one $C_v^2$(Hexagonal) or in other words $C_v^1$(Kagome). Figure 3.6 is the next iteration, a $C_v^3$(Hexagonal) or $C_v^2$(Kagome).

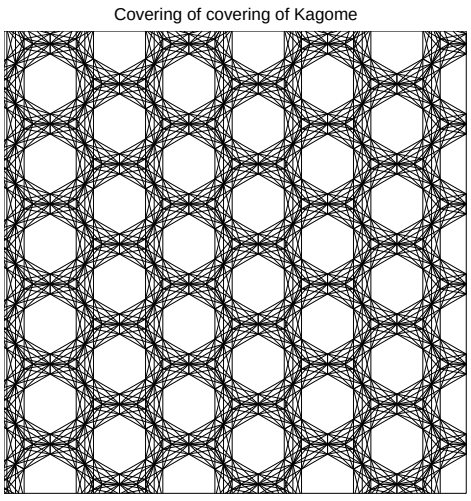



**Figure 3.6** The next covering lattice..

Now let us look at the Voronoi graph and its covering lattices. Pictures in Figure 3.7 are drawn by first creating and cropping a Voronoi graph with the help of the program `crop.m`, then use the recursive procedure described above to find up to the third covering lattice.

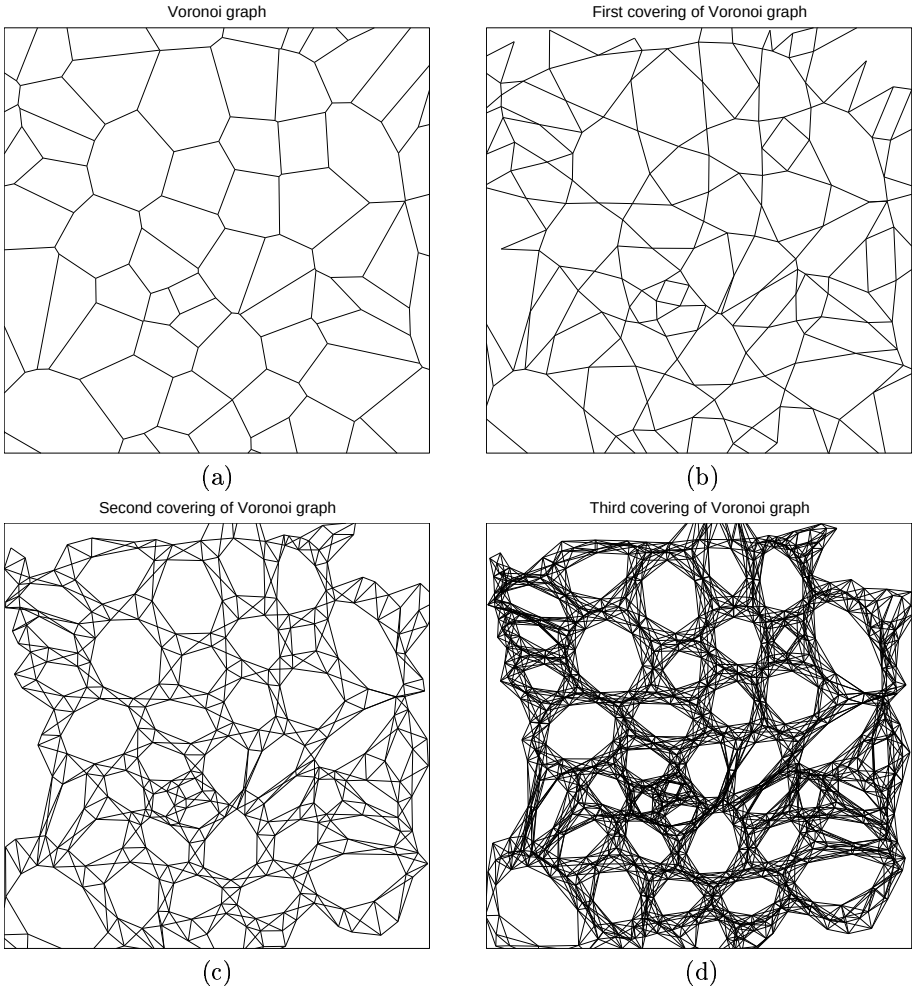



**Figure 3.7** (a) *Voronoi graph (V.g.),* (b) $C_v^1$(**V.g.**), (c) $C_v^2$(**V.g.**), (d) $C_v^3$(**V.g.**).

Notice that the covering lattices retain the skeleton structure of the original Voronoi graph. Those of higher orders represent closer the structures of nature where walls have thickness.

## § 3.1 Quadratic equations

Quadratic equations are equations of binary quadratic forms. Around 400 BC Barbilonia had algorithmic equivalences of quadratic equations which are based on the method of completing the square and where all answers are unsigned, *i.e.* positive, lengths. Because there was no notion for zero, Diophantus considered three types of quadratic equations $ax^2 + bx = c$, $ax^2 = bx + c$, and $ax^2 + c = bx$. Euclid, *circa* 300 BC, used geometric equivalences or quadratic equations whose roots are also lenghts. Brahmagupta allowed negative quantities, which he called debts, and used abreviations for the unknown. Al-Khwarizmi classified quadratics into six types, namely squares equals roots, squares equals numbers, roots equal numbers, squares and roots equal number, squares and numbers equal roots, and roots and numbers equal squares. In his book *Liber embadorum*, published in 1145, Abraham bar Hiyya Ha-Nasi (aka Savasorda) gives the complete solution of quadratic equations. Luca Pacioli published *Summa de arithmetica, geometrica, proportioni et proportionalita* (or Summa) in 1494. He also applied quadratic methods to quartics of the form $x^4 = a + bx^2$. Scipione del Ferro solved the cubic equations of the form $x^3 + mx = n$.

## § 3.2 Quadratic forms

The theory of quadratic forms and the theory of matrix are inseparable though the history of these two subjects are somewhat fragmentary. A bilinear form in the sets $x_i$ and $y_i$, $i = 1 \ldots n$, is $\sum_{i,j} x_i y_j$, or $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{y}$. If $x_i = y_i$ for all $i$, then the form is quadratic in $x_i$. In other words, a quadratic form is a general expression which contains second order terms.

A quadratic form in the variables $x_i$ is the homogeneous quadratic polynomial $\sum_{i,j=1}^{n} a_{ij} x_i x_j$, where $a_{ij}$ are arbitrary scalars. The set of all quadratic forms in $x_i$ with coefficients in a field $\mathcal{F}$ is a vector space over $\mathcal{F}$. A bilinear form in the variables $x_i$ and $y_i$ is the homogeneous polynomial $\sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij} x_i y_j$, where $a_{ij}$ are arbitrary scalars. The set of all bilinear forms with coefficients in a field $\mathcal{F}$ is a vector space over $\mathcal{F}$ (*cf* Hohn, 1958). A quadratic form can also be represented in a matrix form as $X^{\mathrm{T}} A X$. Distinct $n \times n$ matrices $A_1$ and $A_2$ have the same quadratic polynomial if all corresponding $a_{ij} + a_{ji}$ are equal. Given the quadratic form, it is not possible to identify the corresponding matrix $A$ by inspection. This ambiguity is eliminated by replacing each pair of the coefficients $a_{ij}$ and $a_{ji}$ by their mean $(a_{ij}+a_{ji})/2$, which amounts to replacing $A$ by $(A+A^{\mathrm{T}})/2$ which is symmetric. If $x_i$ are independent variables, the rank and the determinant of $A$ are respectively the rank and the discriminant of the form. Two quadratic forms $X^{\mathrm{T}} A_1 X$ and $\tilde{X}^{\mathrm{T}} A_2 \tilde{X}$ are equivalent if and only if there is a nonsinguar transformation $X = B\tilde{X}$ such that $X^{\mathrm{T}} A_1 X = \tilde{X}^{\mathrm{T}} B^{\mathrm{T}} A_1 B \tilde{X} = \tilde{X}^{\mathrm{T}} A_2 \tilde{X}$, that is if and only if for a suitable nonsingular matrix $B$, $A_2 = B^{\mathrm{T}} A_1 B$. Then $A_1$ and $A_2$ have the same rank and are said to be congruent to each other.

Lattice can be represented by quadratic forms and vice versa, therefore the classification of quadratic forms is also the classification of lattices.

## § 3.3 Voronoi algorithms

Green and Sibson (1978) gives the algorithm that produces Voronoi graphs, as summarised in Algorithm 3.1. They call a Voronoi tessellation 'Dirichlet tessellation'. The points outside the window are ignored, and points on the periphery has tile bounded in part by sequences of effective constraints instead of by inter-tile edges. Contiguous tiles have a boundary sequences in common. A degenerate vertex, that is a vertex where four or more tiles meet, causes an incorrect record of diagonal contiguity. From the Euler-Poincaré formula, $f - e + v = 2$, where in the plane the infinite region is counted as a face, gives rise to the total number of contiguities to be recorded $4\times$ *effective constrains* $+6\times$ *accepted points* $-6$. A square lattice has all its vertices degenerate.

**Algorithm 3.1** *Voronoi 2-d algorithm, Green and Sibson (1978)*

**define** window as a set of linear inequality constraints $ax + by + c < 0$;
**do**
  **add** a new point $n$ randomly within the window;
  **do**
    **find** the line $b_i$ joining $n$ and its nearest neighbour $p_i$;
    **find** $l_i$ the perpendicular bisector of $b_i$;
    **find** the intersection between $l_i$ and the edge $e_i$ of its nearest tile,
      going clockwise with respect to $n$;
    **add** $l_1$ to the object- and contiguity lists;
  **until** $2\pi$ radian around $n$ traversed
**until** no more new points added **enddo** □

Moore and Angell developed a algorithm which can cope with degenerate cases. Because their program could find them, they are no longer considered degenerate and they fittingly called these *vertices of >3-hedral valency.*

## § 3.4 Voronoi statistics in literature

Van de Weygaert (1994) studies linear section of three dimensional Voronoi network where the mean length is

$$\langle\lambda\rangle = L = \frac{d\Gamma(d-1/2)\Gamma((d+1)/2)^2}{(d-1)!\rho_c^{1/d}2\Gamma(d/2+1)^{2-(1/d)}\Gamma(2-1/d)}. \qquad (\mathbf{12})_{\text{iii}}$$

For two dimensions $\langle\lambda\rangle = \pi/(4\sqrt{\rho})$, while for three dimensions $\langle\lambda\rangle = [81/(32\pi\rho)]^{1/3}/\Gamma(2/3)$. Consequently for two dimensions $\langle\lambda\rangle = 0.7854\rho^{-1/2}$ while for three dimensions $\langle\lambda\rangle = 0.6872\rho^{-1/3}$, $\langle\lambda^2\rangle = 0.632\rho^{-2/3}$, $0.668\rho^{-1}$ and $\langle\lambda^4\rangle = 0.774\rho^{-4/3}$. For 3-d Poisson-Voronoi tessellation the exact values for the moments are shown in Table 3.1.

| | | | | |
|---|---|---|---|---|
| $\rho_v$ | $\frac{(24}{35)\pi^2\rho_c}$ | $6.763\rho_c$ | number density of vertices | $J_0$ |
| $\rho_e$ | $\frac{48}{35}\pi^2\rho_c$ | $13.535\rho_c$ | number density of edges | $J_1$ |
| $\rho_f$ | $\left(\frac{24}{35}\pi^2+1\right)\rho_c$ | $7.768\rho_c$ | number density of faces | $J_2$ |
| $\rho_c$ | $\rho_c$ | $\rho_c$ | number density of cells | $J_3$ |
| $n_c^v$ | $\frac{96}{35}\pi^2$ | 27.07 | number of vertices per cell | $\mathrm{E}(N_0)$ |
| $n_c^e$ | $\frac{144}{35}\pi^2$ | 40.61 | number of edges per cell | $\mathrm{E}(N_1)$ |
| $n_c^f$ | $\frac{48}{35}\pi^2+2$ | 15.54 | number of faces per cell | $\mathrm{E}(N_2)$ |
| $n_f^v$ or $n_f^e$ | $\frac{144\pi^2}{24\pi^2+35}$ | 5.228 | number of vertices or edges per face | $E_2(N_0)$ or $E_2(N_1)$ |
| $V_c$ | $\frac{1}{\rho_c}$ | $\frac{1}{\rho_c}$ | volume of cell | $\mathrm{E}(V)$ |
| | $\frac{1.180}{\rho_c^2}$ | $\frac{1.180}{\rho_c^2}$ | | $\mathrm{E}(V^2)$ |
| $A_c$ | $\left[\frac{256\pi}{3\rho_c^2}\right]^{1/3}\Gamma\left(\frac{5}{3}\right)$ | $5.821\rho_c^{-2/3}$ | surface area of cell | $\mathrm{E}(A_c)$ |
| $s_c$ | $\frac{(4\pi)^{5/3}\Gamma(1/3)}{5(9\rho_c)^{1/3}}$ | $17.496\rho_c^{-1/3}$ | perimeter of cell | $\mathrm{E}(s_c)$ |
| $A_f$ | $\frac{35(2^{8/3})\pi^{1/3}\Gamma(2/3)}{(9\rho_c)^{2/3}(24\pi^2+35)}$ | $0.3747\rho_c^{-2/3}$ | area of face | $\mathrm{E}(A_w)$ |
| $s_f$ | $\frac{7(2^{10/3})\pi^{5/3}\Gamma(1/3)}{(9\rho_c)^{1/3}(24\pi^2+35)}$ | $2.252\rho_c^{-1/3}$ | perimeter of face | $\mathrm{E}(s_w)$ |
| $d_e$ | $\frac{7\Gamma(1/3)}{9(36\pi\rho_c)^{1/3}}$ | $0.4309\rho_c^{-1/3}$ | length of edge | $\mathrm{E}(L)$ |

**Table 3.1** *Moments of 3-d Poisson-Voronoi tessellation, cf van de Weygaert (1994).*

The moments of 2-d section of 3-d Voronoi tessellation are shown in Table 3.2.

| | | | | |
|---|---|---|---|---|
| $\rho_v$ | $\frac{2\Gamma\left(\frac{1}{3}\right)\left(\frac{16\pi^5\rho^2}{9}\right)^{1/3}}{15}$ | $2.9159\rho_c^{2/3}$ | number density of vertices | $J_0$ |
| $\rho_e$ | $\frac{3}{2}\rho_v$ | $4.3739\rho_c^{2/3}$ | number density of edges | $J_1$ |
| $\rho_c$ | $\frac{1}{2}\rho_v$ | $1.4530\rho^{2/3}$ | number density of cells | $J_2$ |
| $n_c^v$ or $n_c^e$ | 6 | 6 | number of vertices or edges per cell | $\mathrm{E}(N_0)$ or $\mathrm{E}(N_1)$ |
| $A_c$ | $\frac{15}{\Gamma\left(\frac{1}{3}\right)\left(\frac{16\pi^5\rho_c^2}{9}\right)^{1/3}}$ | $0.6859\rho^{-2/3}$ | | $\mathrm{E}(A)$ |
| | | $0.698\rho_c^{-4/3}$ | | $\mathrm{E}(A^2)$ |
| $s_c$ | $\frac{30\Gamma\left(\frac{2}{3}\right)}{(36\pi\rho_c)^{1/3}\Gamma\left(\frac{1}{3}\right)}$ | $3.1356\rho_c^{-1/3}$ | perimeter of cell | $\mathrm{E}(S)$ |
| $d_e$ | $\frac{5\Gamma\left(\frac{2}{3}\right)}{(36\pi\rho_c)^{1/3}\Gamma\left(\frac{1}{3}\right)}$ | $0.5226\rho_c^{-1/3}$ | length of edge | $\mathrm{E}(L)$ |

**Table 3.2** *Moments of 2-d section of 3-d Voronoi, cf van de Weygaert (1994).*

The form factor of a cell, $f_c = 36\pi\frac{V_c^2}{A_c^3}$, is a dimensionless parameter which partially describes the shape of the cell. The form factor of a face is $f_f = 4\pi\frac{A_f}{s_f^2}$, which is the unity when the face is a circle. Regularly shaped cells have their shape approaching that of the sphere where $f_f = 1$. Statistics of 3-d VT which van de Weygaert (1994) gives include those listed here in Table 3.3.

| | $\mu(\cdot)$ | $\sigma(\cdot)$ | $\gamma_1(\cdot)$ | $\gamma_2(\cdot)$ | |
|---|---|---|---|---|---|
| $\rho_v$ | $6.747 \pm 0.014$ | | | | $J_0$ |
| $\rho_e$ | $13.493 \pm 0.028$ | | | | $J_1$ |
| $\rho_f$ | $7.747 \pm 0.014$ | | | | $J_2$ |
| $\rho_c$ | $1.000 \pm 0.000$ | | | | $\rho$ |
| $n_c^v$ | $26.986 \pm 0.055$ | $6.61 \pm 0.18$ | $0.346 \pm 0.029$ | $-0.029 \pm 0.030$ | $N_0$ |
| $n_c^e$ | $40.479 \pm 0.083$ | $9.92 \pm 0.27$ | $0.346 \pm 0.029$ | $-0.029 \pm 0.030$ | $N_1$ |
| $n_c^f$ | $15.493 \pm 0.028$ | $3.305 \pm 0.091$ | $0.346 \pm 0.029$ | $-0.029 \pm 0.030$ | $N_2$ |
| $V_c$ | $1.000 \pm 0.000$ | $0.418 \pm 0.009$ | $0.73 \pm 0.11$ | $0.70 \pm 0.39$ | $V_{cell}$ |
| $A_c$ | $5.801 \pm 0.018$ | $1.461 \pm 0.043$ | $0.28 \pm 0.11$ | $-0.01 \pm 0.17$ | $A_{cell}$ |
| $s_c$ | $17.443 \pm 0.054$ | $3.655 \pm 0.094$ | $0.305 \pm 0.035$ | $0.00 \pm 0.13$ | $S_{cell}$ |
| $f_c$ | $0.540 \pm 0.006$ | $0.082 \pm 0.003$ | $-0.565 \pm 0.032$ | $0.36 \pm 0.11$ | $F_{cell}$ |
| $n_f^v$ | $5.2255 \pm 0.0014$ | $1.564 \pm 0.016$ | $0.582 \pm 0.016$ | $0.058 \pm 0.024$ | $N_0^w$ |
| $A_f$ | $0.3744 \pm 0.0015$ | $0.3722 \pm 0.0037$ | $1.266 \pm 0.047$ | $1.40 \pm 0.19$ | $A_{wall}$ |
| $s_f$ | $2.2518 \pm 0.0092$ | $1.2009 \pm 0.0050$ | $0.089 \pm 0.022$ | $-0.796 \pm 0.029$ | $S_{wall}$ |
| $f_f$ | $0.6389 \pm 0.0005$ | $0.1635 \pm 0.0011$ | $-0.855 \pm 0.035$ | $0.40 \pm 0.13$ | $F_{wall}$ |
| $d_{cf}$ | $0.6402 \pm 0.0017$ | $0.2092 \pm 0.0051$ | $-0.024 \pm 0.022$ | $-0.333 \pm 0.074$ | $D_{nw}$ |
| $V_{cf}$ | $0.0645 \pm 0.0001$ | $0.0579 \pm 0.0009$ | $1.177 \pm 0.097$ | $1.52 \pm 0.47$ | $V_{nw}$ |
| $d_e$ | $0.4309 \pm 0.0018$ | $0.3216 \pm 0.0023$ | $0.829 \pm 0.020$ | $0.209 \pm 0.044$ | $L$ |
| $\alpha_{ee}$ | 111°C107 $\pm$ 0.018 | 35°C310 $\pm$ 0.078 | $-0.499 \pm 0.012$ | $-0.276 \pm 0.039$ | $\alpha_{ee}$ |
| $\alpha_{ff}$ | 120°C0 $\pm$ 0.0 | 23°C53 $\pm$ 0.30 | $-0.296 \pm 0.014$ | $-0.255 \pm 0.038$ | $\alpha_{ww}$ |

**Table 3.3** *Statistics of 3-d VT, van de Weygaert (1994)*

His statistics on the planar section of the 3-d Voronoi tessellation is shown here as Table 3.4.

| | | | | |
|---|---|---|---|---|
| $\rho_s^c$ | 1.4530 | 0.0592 | 0.2871 | -1.0099 |
| $\rho_s^v$ | 2.9060 | 0.1134 | 0.2871 | -1.0099 |
| $\rho_s^e$ | 4.3590 | 0.1776 | 0.2871 | -1.0099 |
| $n_s^v$ | 6.0000 | 1.6895 | 0.3311 | -0.1142 |
| $A_s^c$ | 0.6882 | 0.4741 | 0.4573 | -0.3902 |
| $s_s^c$ | 3.1418 | 1.2212 | -0.5819 | -0.2879 |
| $f_s^c$ | 0.7050 | 0.1433 | -1.3425 | 2.0875 |
| $d_s^e$ | 0.5221 | 0.3631 | 0.6223 | -0.2321 |
| $\alpha_s^{ee}$ | 120°C0000 | 31°C6039 | -0.6351 | 0.1321 |
| $\alpha_s^{ff}$ | 63°C1812 | 18°C5232 | -0.6351 | -0.3520 |

**Table 3.4** *Statistics of the planar section of the 3-d Voronoi tessellation, van de Weygaert (1994).*

Then Table 3.5 lists the statistics he gave for the line section of 3-d VT.

| | | | | |
|---|---|---|---|---|
| $\lambda$ | 0.6703 | 0.3942 | 0.2600 | -0.6923 |
| $\sigma(\lambda)$ | 0.6123 | 0.1159 | 0.0643 | 0.0204 |
| $\alpha_l^{fl}$ | 44°C69 | 19°C5325 | 0.0067 | -0.8080 |
| $\alpha_l^{ff}$ | 75°C4123 | 30°C9389 | -0.0185 | -0.5829 |

**Table 3.5** *Statistics for the line section of the 3-d Voronoi tessellation, van de Weygaert (1994).*

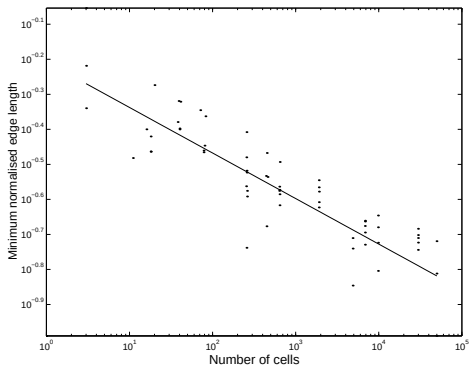
Tables 3.5, 3.5 and 3.5 are all on VT's which are created from $\mathcal{P}(0,3)$. Apart from these, van de Weygaert (*ibid.*) also gives results based on VT's with anticorrelated- and correlated nuclei the statistics of which are not listed here.

Another interesting statistics is the correlation between the number of sides of a grain $n_e$, and the expected number of sides of its neighbouring grains $n_n^e$. Aboav (1970), in his investigation of polycrystalline MgO, empirically found it to be $n_n^e = 5 + 8/n_e$ and Weaire (1974) found a theoretical value of $n_n^e = 5 + 6/n_e$.

Later Aboav (1983) published another finding in which he studied a thin film of arsenic-selenium glass, $As_2Se_3$. This time with the initial assumption that $n_n^e = 6 - a + (6a + \mu_2)/n$ and $\mu_2 = \sum_n (n-6)^2 f_n = \langle (n-6)^2 \rangle$ he empirically finds $n_n^2 = 4.79 + 8.98/n$.

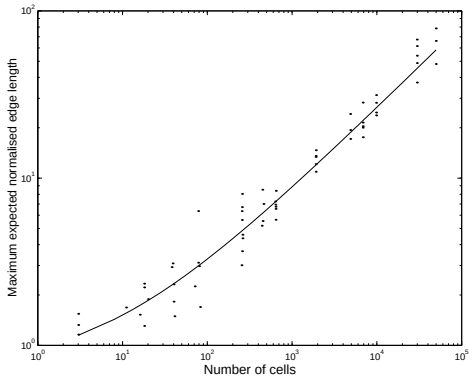
## § 3.5 Voronoi statistics

In two dimensions the statistical descriptions are as follows.



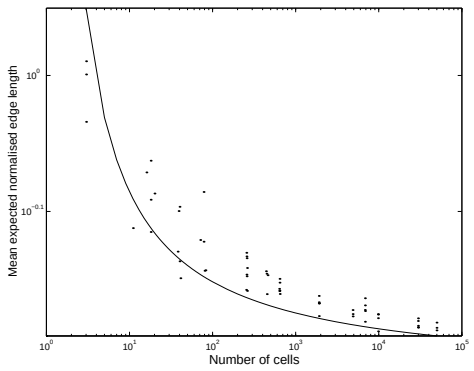

The curve in Figure 3.8 is $y = 0.62/n^{0.13}$. First the edge lengths are normalised by the edge length of the equivalent or characteristic square . A *equivalent square* is defined as the square figure whose area is equal to the area of the polygon in question, here a Voronoi polygon. Then find the average of the edge lenghts in each cell.

**Figure 3.8** The minimum of the average normalised edge length.




In Figure 3.9 the curve is $y = \left|(n/15)^{1/2}\right| + 0.7$. Of these cell-averaged normalised edge length obtained from simulations on various sizes of networks the minimum values are plotted in Figure 3.8, the maximum in Figure 3.9, and the expected value in Figure 3.10. Note that the last quantity is the average over the whole structure of all the averages obtained one from each cell.

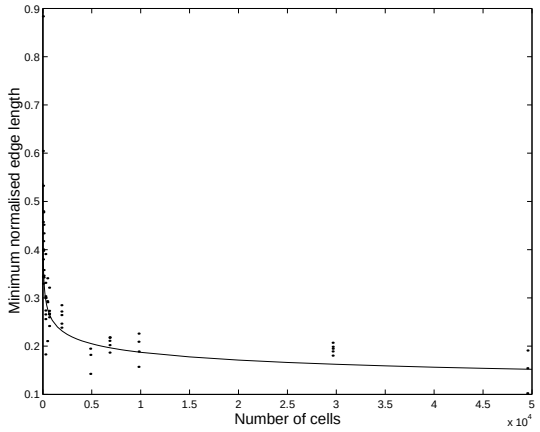
**Figure 3.9** The maximum of the average normalised edge length..



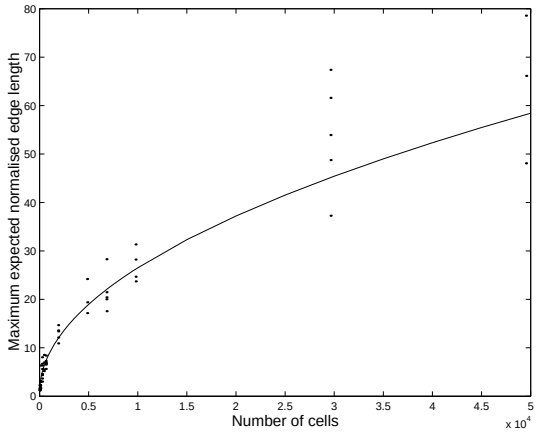

In Figure 3.10 the curve is $y = 10^{0.2/\log x} - 0.4$. To summarise, as the networks gets larger its minimum, maximum, and mean of the edge lenghts when compared with the characteristic length approach constant values. The characteristic length is defined as $l = [\sum_{i=1}^{n} A_i/n]^{1/2}$, where $A_i$ is the area of the $i^{\text{th}}$ polygon and $n$ is the number of cells.

**Figure 3.10** The mean of the average normalised edge length..

That the three values mentioned become constant may not seem obvious by the look of Figures 3.8 to ℵ because the scale used there is a logarithmo-logarithmic scale, not a Euclidean one. These figures emphasise the smaller ranges of size. Figure 3.11 below on the other hand is plotted using a normal scale which enables one to see the asymptotic effect more clearly. Here the figures (a), (b), and (c) are respectively Figure 3.8, 3.9, and *manel*. Let the term *representative* stands for 'of the cell-average normalised', and *length* means 'edge length' in this context. Then the *minimum representative length* approaches 0.15 from Figure 3.11 (a), the *maximum representative length* is ever increasing, seemingly by a power law of approximately 0.5, while *mean representative length* approaches the value of 0.65. Properties of the Voronoi tessellation can be divided into individual and collective properties. With this in mind the term *length* above represents a property, *representative* means individual, and *minimum*, *maximum* and *mean* show the collective attributes.




(a)




(b)

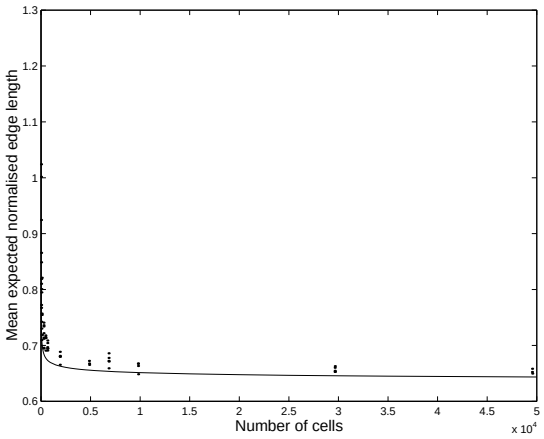
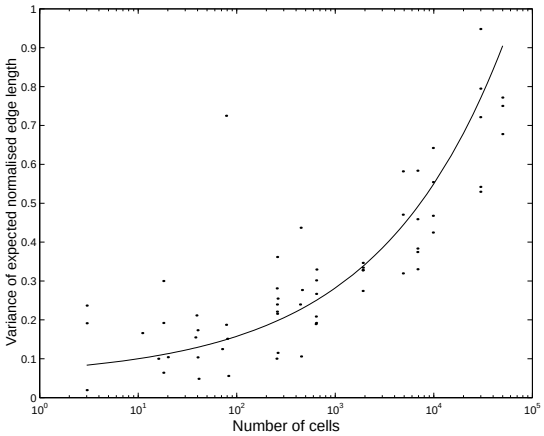
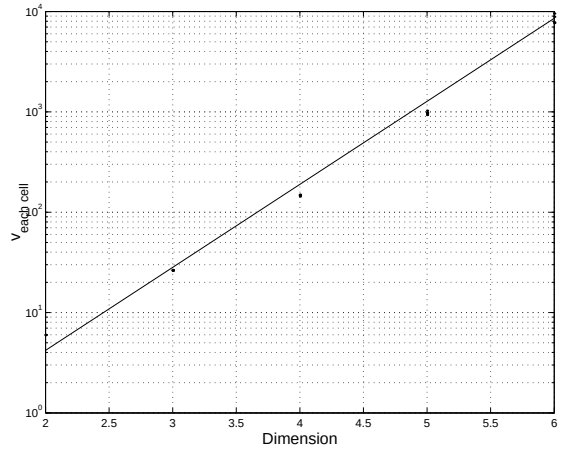



**Figure 3.11** Mean, maximum, and mean of the cell-average normalised edge length.



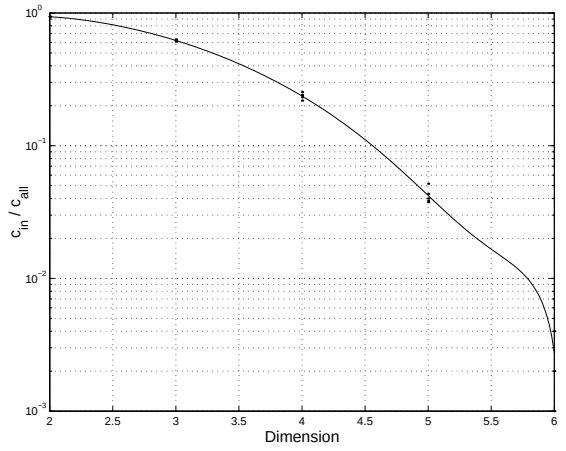

This $\sigma^2(\mathcal{N}_v^c(E(l_e)))$ increases very slowly with the increasing sizes of the networks. The curve shown has the equation $y = \left|\left[x/(8 \times 10^4)\right]^{1/3}\right| + 0.05$.

**Figure 3.12** The variance of the expected values of the normalised length of edges of a cell.




The number of vertices per cell increases dramatically as one goes up the dimension ladder. The program in § A.10 contains the essential part of the code which produces Figure 3.13. The straight line shown is $0.093(4+e)^n$. Notice the trend towards a greater rate of increase at dimensions higher than the maximum six shown. Only Voronoi cells which lies within the original domain and the vertices of these are considered. The same program also gives Figure 3.14.

**Figure 3.13** Number of vertices per cell..




The expanse and the hypervolume of the Voronoi tessellation increases at an enormous rate, which results in the number of cells totally bounded within the original domain decreasing rapidly in Figure 3.14 as the dimension goes up. The effect at close to the zero ratio is emphasised by using the log scale for the $y$-axis. Also with the logarithmic scale the polynomial estimation curves that give negative values of the ratio is automatically excluded.

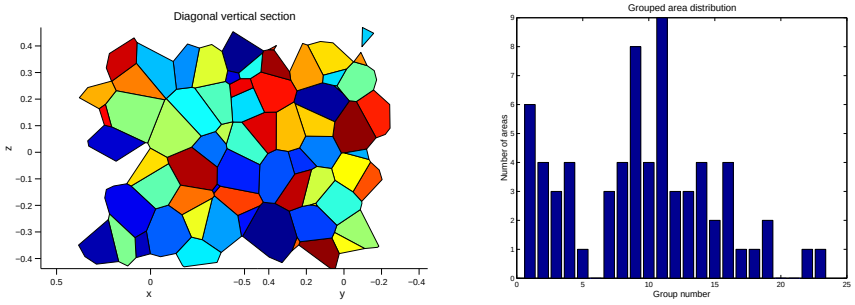
**Figure 3.14** Ratio of cells in the original domain..

One can fit the polynomial $p(x) = p_1x^n + p_2x^{n-1} + \ldots + p_nx + p_{n+1}$ to the data with a least square algorithm. If $\mathbf{x}$ is the vector containing the data, then the the $(n+1)$ coefficients of the estimated polynomial can be found from $\hat{\mathbf{x}} = (\mathbf{x} - \mathrm{E}(\mathbf{x}))/\sigma(\mathbf{x})$. For data containing independent normal errors with a constant variance, the error bounds contain at least half of the predictions. The curve shown in Figure 3.14 is $p(x) = -0.048x^4 + 0.064x^3 - 0.203x^2 - 0.459x + 0.263$, the average value $\mathrm{E}(\mathbf{x})$ is 3.917, and the standard deviation $\sigma(\mathbf{x})$ is 1.412. The structure of the polynomial fit can be described using the Cholesky factor of the Vandermonde matrix

$$R = \begin{bmatrix} -12.19 & -1.56 & -6.08 & -0.47 & -3.17 \\ 0 & 8.46 & -0.45 & 4.47 & -0.44 \\ 0 & 0 & 1.21 & 0.33 & 2.93 \\ 0 & 0 & 0 & -1.63 & 0.30 \\ 0 & 0 & 0 & 0 & 2.25 \end{bmatrix},$$

the degree of freedom which is 19, and the norm of the residuals which is 0.034 in this case.

## § 3.6 Voronoi section




(a) (b)

**Figure 3.15** *(a) Section by the plane $x - y + \epsilon z = \epsilon, \epsilon = 10^{-4}$. (b) Grouped distribution of area, the number of groups is approximately one-third the number of regions.*

| | $\mathcal{V}_c$ | $\aleph_c$ | $A_c^{fr}$ | $V_c^{fr}$ | $n_c^e$ | $E(n_{f,c}^e)$ | $\sigma(n_{f,c}^e)$ |
|---|---|---|---|---|---|---|---|
| min | 8 | 6 | $2.2191 \times 10^{-4}$ | $9.9722 \times 10^{-5}$ | 12 | 4 | 0.6030 |
| max | 46 | 25 | $4.7015 \times 10^{-3}$ | 0.14364 | 69 | 5.52 | 2.5690 |
| $\mu$ | 26.338 | 15.169 | $1.8975 \times 10^{-3}$ | $1.8975 \times 10^{-3}$ | 39.507 | 5.1712 | 1.5586 |
| $\sigma^2$ | 40.608 | 10.152 | $5.2144 \times 10^{-7}$ | $4.2313 \times 10^{-5}$ | 91.368 | 0.035983 | 0.1186 |
| $\sigma$ | 6.3725 | 3.1862 | $7.2210 \times 10^{-4}$ | $6.5048 \times 10^{-3}$ | 9.5587 | 0.18969 | 0.3444 |
| $\mu_g$ | 25.543 | 14.829 | $1.7572 \times 10^{-3}$ | $1.1575 \times 10^{-3}$ | 38.315 | 5.1676 | 1.5169 |
| $\mu_h$ | 24.703 | 14.478 | $1.6026 \times 10^{-3}$ | $8.5548 \times 10^{-4}$ | 37.054 | 5.1638 | 1.4700 |
| med | 26 | 15 | $1.8125 \times 10^{-3}$ | $1.1660 \times 10^{-3}$ | 39 | 5.2 | 1.5706 |
| mad | 5.0068 | 2.5034 | $5.6975 \times 10^{-4}$ | $1.4250 \times 10^{-3}$ | 7.5103 | 0.14465 | 0.2715 |
| $\mathcal{M}^2$ | 40.531 | 10.133 | $5.2045 \times 10^{-7}$ | $4.2233 \times 10^{-5}$ | 91.195 | 0.035915 | 0.1184 |
| $\mathcal{M}^3$ | 72 | 9 | $2.5644 \times 10^{-10}$ | $5.4435 \times 10^{-6}$ | 243 | $-8.3404 \times 10^{-3}$ | -0.0060 |
| $\mathcal{M}^4$ | 5088.9 | 318.06 | $1.0467 \times 10^{-12}$ | $7.6653 \times 10^{-7}$ | 25763 | $7.9566 \times 10^{-3}$ | 0.041906 |
| $\mathcal{K}$ | 3.0978 | 3.0978 | 3.8644 | 429.77 | 3.0978 | 6.1689 | 2.9916 |

**Table 3.6** *Simulation uses rbox (1000 random points, seed 234985) and qhull (option v and o); $d = 3$, $n^c = 527$, $n^v = 6357$, CPU time 6,466.99 sec for the counting of statistics, 270.56 sec for finding area of the faces, 4.47 sec for calculating cell volume and 2.8 sec for finding the number of edges.*

## § 3.7 Number of vertices and edges

It has been observed from the simulations that in three dimensions cells always have vertices in even numbers and edges odd ones. This can be explained by the following theorems.

**Theorem 3.2.** *(cf Miles, 1972) In a simple three dimensional Voronoi tessellation, $3n_c^v = 2n_c^e$.*

**Proof.:** Pick any Voronoi cell of the tessellation. Suppose that it has $n^v$ vertices. Add up the number of edges connected to all vertices. Because every cell is a simple polyhedron, there are exactly three edges connected to each vertex. The number of edges thus counted is therefore $3n^v$. But each of the edges is connected to two vertices, so we have counted every one of them twice. Therefore,

$$2n^e = 3n^v.$$

This is the case for any cell, hence the theorem is proved. □

This theorem gives rise the following two theorems.

**Theorem 3.3.** *The number of vertices of any cell within a simple three dimensional Voronoi tessellation is an even positive integer.*

**Proof.:** Observe that the term $2n_c^e$ in the theorem above is divisible by 2. This term is equal to $3n_c^v$, therefore the latter is also divisible by 2. Since 2 can not divide into 3, the only term left, $n_c^v$, must be divisible by 2 and hence even number. □

**Theorem 3.4.** *The number of edges of any cell within a simple three dimensional Voronoi tessellation is a positive integer divisible by three.*

**Proof.:** With the same line of reasoning as above, observe that $3n_c^v$ is divisible by 3. Therefore $2n_c^e$, and hence $n_c^e$, is also divisible by 3. □

Another proof for both the above theorems is the following.

**Proof.:** For an equality to hold, both sides must have the same factors. By supposing an unknown common factor $i$ and by cross-multiplying the coefficients on both sides, one obtains $2\cdot(3\cdot i) = 3\cdot(2\cdot i)$, where $i$ is a positive integer. Therefore $n_c^e = 3i$ and $n_c^v = 2i$. In other words, $n_c^e$ is divisible by three and $n_c^v$ is even. □

The theorems above assume that every cells are simple. This can not be the case in real situation where edges have dimensions and rather represent tubes than one-dimensional lines. Such case is similar to the so-called *degenerative* case in a computational model of Voronoi tessellation where there exist vertices the number of edges connected to each of which exceeds four. Even in the degenerative case, one would perhaps still expect a tendency for $n_c^v$ to be an even number and for $n_c^e$ to be divisible by three to hold.

In nature there are things which have a tendency towards even numbers. The following graph shows the the abundance in cosmic materials from the compilation by Cameron (1973). The year of publication of this paper is often misquoted as 1970, due to misprints in a footnote on the first page of the paper. Even the legendary Fred Hoyle has consistently practised this mistake and let it go uncorrected throughout his career. From what I have come across without an effort of searching, in two of his books and at least one of his papers, spanning the period of roughly fourty years in total (*cf* Hoyle, 1977). Out of a sample of 278 of those papers which cite this work, this misprint resulted in 80% of the total number of errors in the year cited, which in turn amounts to 1.8% of the number of samples.

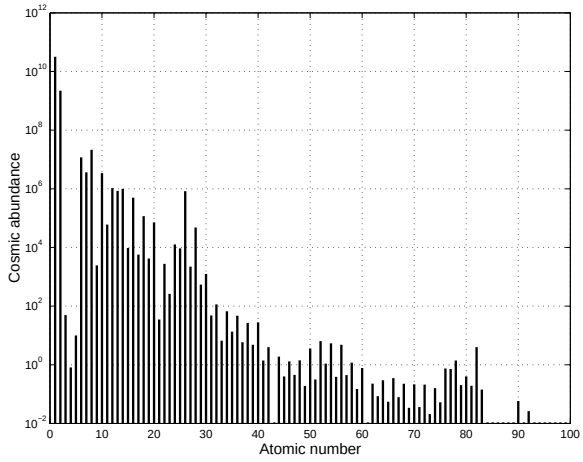
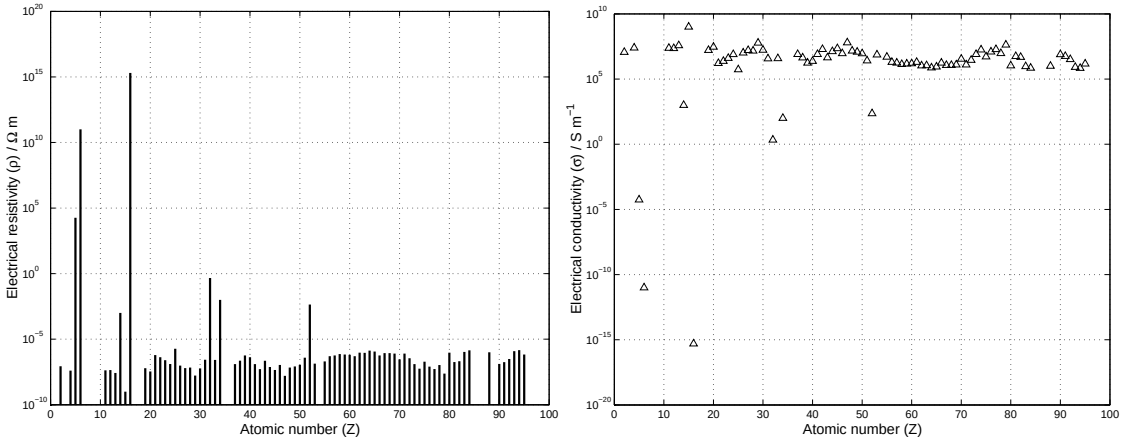



**Figure** ℵ fgabn

*The abundances of the chemical elements* in the universe. They are assumed to be the same as those found in the primitive solar nebula, which have been deduced from data on abundances found in chondritic meteorites and those found in the Sun. All abundances are relative to that of Si which is taken to be $10^6$. Missing bars appear where the atomic numbers are unstable. Except for the atomic number 1 of Hydrogen, which is the most universal element, all other elements with even atomic numbers are locally more abundant than those with near-by odd atomic numbers.:

Of interest are also the electrical resistivity and conductivity of solid matters.The conductivity $\sigma$ is by definition the reciprocal of the resistivity $\rho$. Some of the solids, particularly boron, carbon, silicon, sulphur, germanium, selenium and tellurium, have a distinctively higher resistivity than the majority. Interestingly all of these, with only one exception of boron whose atomic number is five, are of an even atomic number, which respectively from carbon are 6, 14, 16, 32, 34, and 52. This can be seen in Figure 3.17 the data of which are taken from Podesta (2002).




(a) (b)

**Figure 3.17** *The electrical resistivity, (a), and conductivity, (b), of elements which are solid at the room temperature.*

| | $n_c^e$ | $\mathcal{N}_v^c(E(l_c^e))$ | $\mathcal{N}_v^c(\wp_c)$ | $\mathcal{N}_v^c(A_c)$ |
|---|---|---|---|---|
| min | 3 | 0.20716 | 0.2749 | 0.058428 |
| max | 11 | 8.1072 | 10.134 | 22.307 |
| $\mu$ | 5.8973 | 0.70626 | 1.0089 | 1 |
| $\sigma^2$ | 1.8955 | 0.18607 | 0.2942 | 1.4379 |
| $\sigma$ | 1.3768 | 0.43136 | 0.54241 | 1.1991 |
| $\mu_g$ | 5.742 | 0.66036 | 0.94795 | 0.79225 |
| $\mu_h$ | 5.5904 | 0.62941 | 0.89999 | 0.61799 |
| med | 6 | 0.65709 | 0.96614 | 0.84267 |
| mad | 1.0736 | 0.17995 | 0.24357 | 0.49216 |
| $\mathcal{M}^2$ | 1.8912 | 0.18565 | 0.29355 | 1.4347 |
| $\mathcal{M}^3$ | 1.6194 | 0.96371 | 1.7931 | 22.703 |
| $\mathcal{M}^4$ | 12.438 | 6.8376 | 15.785 | 468.03 |
| $\mathcal{K}$ | 3.4775 | 198.38 | 183.18 | 227.39 |

**Table 3.7** *Neighbour statistics. Here for the normalisation purpose,* $l^e_{\text{basis}} = 0.046662$, $\wp_{\text{basis}} = 0.18665$, $A_{\text{basis}} = 0.0021773$. *Simulation uses voronoin command in Matlab;* $d = 2$, $n^c = 448$, $n^v = 946$, *CPU time 1.19 seconds.*

Next simulation was done with $d = 2$, $n^c = 3$ to 49551.

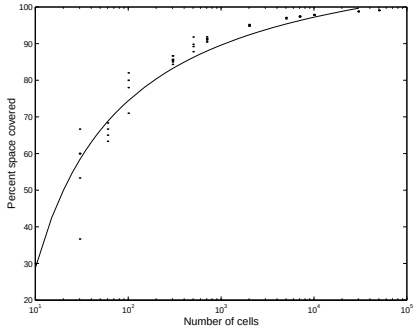
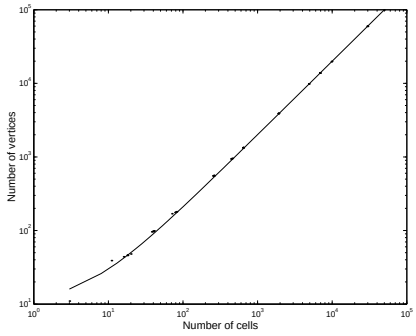
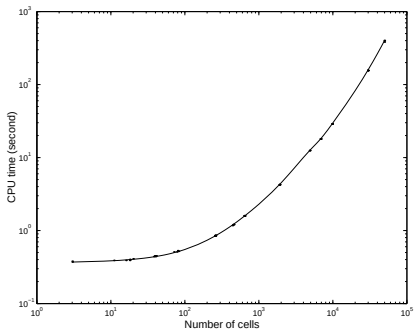



**Figure 3.18** Percentage of space covered by a Voronoi structure. The number of cells is the total number of cells generated. The percent space covered is the volume of the structure after boundary cells, that is cells which extrude the unit volume boundary, have been excluded. The equation of the reference curve is $y = -210/\log x + 120$.



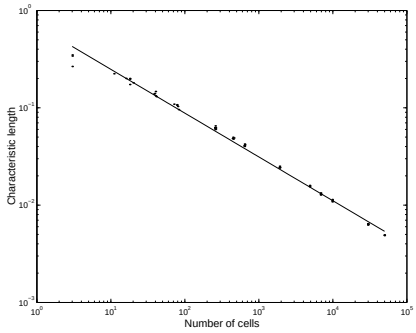

The reference line in Figure 3.20 is the linear equation $y = 2x + 10$. Boundary cells have been excluded.

**Figure 3.20** Number of vertices versus number of cells..



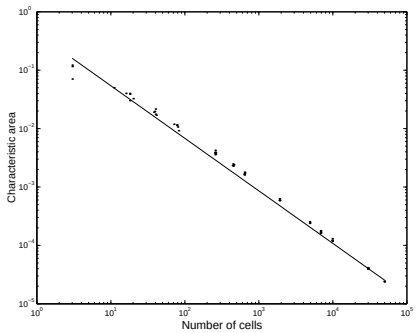

The curve in Figure 3.22 is the result of curve fitting by cubic spline interpolation.

**Figure 3.22** The CPU time in seconds..




The characteristic length is the length of the side of the cubic structure having the same number of cells and the same total volume as the Voronoi structure. The characteristic lenghts in Figure 3.24 are shown as dots. The reference line is $y = 0.7/x^{0.45}$.

**Figure 3.24** Characteristic length versus the number of cells..

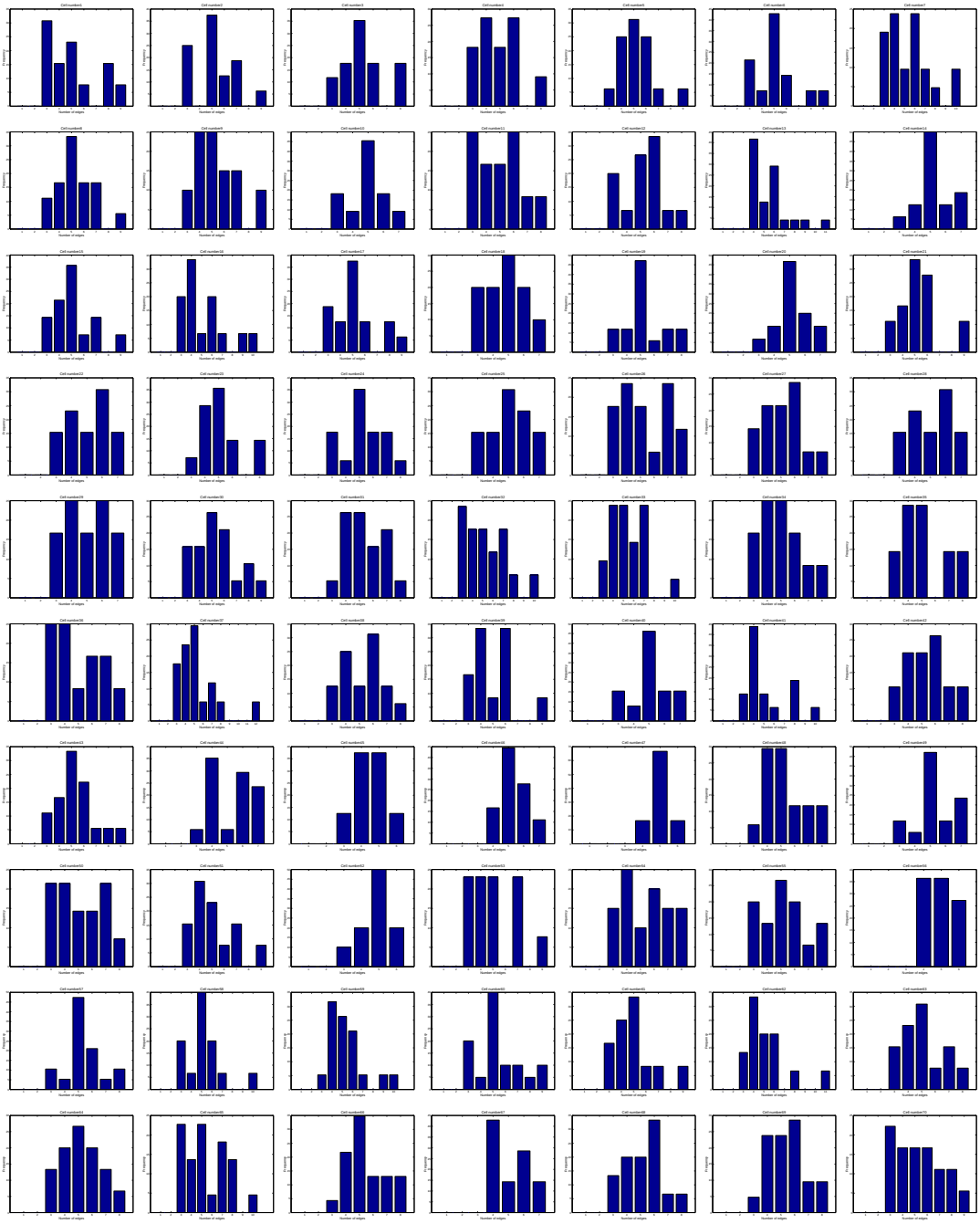


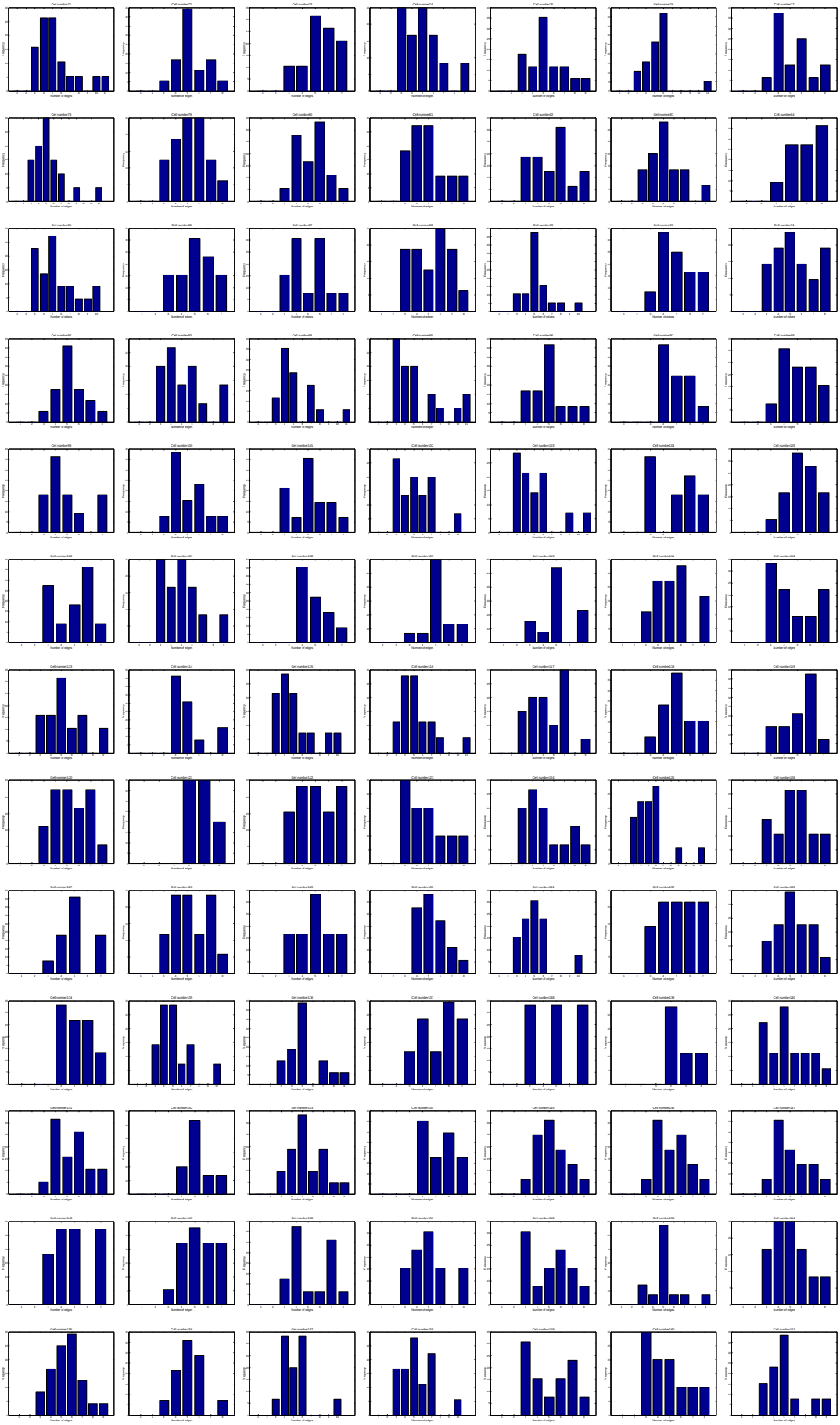

The characteristic area is the area of each square in the assembly the total volume and the number of cells of which are the same as those of the Voronoi graph. The reference curve shown in Figure 3.26 is $y = 0.43/x^{0.9}$.

**Figure 3.26** Characteristic area versus the number of cells..

| | $n_c^v$ | $n_f^v$ | $\mathcal{N}_v^c(\mathcal{P}_c^f)$ | $\mathcal{N}_v^c(A_f)$ | $\mathcal{N}_v^c(A_c)$ | $\mathcal{N}_v^c(V_c)$ | $n_c^f$ |
|---|---|---|---|---|---|---|---|
| min | 8 | 3 | 0.001414 | $4.6179 \times 10^{-7}$ | 0.11908 | 0.21464 | 6 |
| max | 42 | 10 | 0.84885 | 0.82919 | 0.50425 | 12.719 | 23 |
| $\mu$ | 24.423 | 5.1114 | 0.32876 | 0.12535 | 0.30315 | 1.0000 | 14.211 |
| $\sigma^2$ | 44.105 | 2.1650 | 0.030844 | 0.015846 | 0.008511 | 2.5764 | 11.026 |
| $\mu_g$ | 23.504 | 4.9065 | 0.25897 | 0.051216 | 0.28815 | 0.70118 | 13.825 |
| $\mu_h$ | 22.506 | 4.7086 | 0.12620 | $2.50 \times 10^{-4}$ | 0.27198 | 0.58794 | 13.423 |
| med | 22 | 5 | 0.33219 | 0.084165 | 0.30368 | 0.64077 | 13 |
| mad | 5.4338 | 1.1655 | 0.14499 | 0.099433 | 0.075445 | 0.67950 | 2.7169 |
| $\mathcal{M}^2$ | 43.483 | 2.1619 | 0.030800 | 0.015823 | 0.008391 | 2.5401 | 10.871 |
| $\mathcal{M}^3$ | 112.13 | 1.8902 | $5.92 \times 10^{-4}$ | 0.002805 | $4.8 \times 10^{-5}$ | 24.169 | 14.016 |
| $\mathcal{K}$ | 3.1096 | 3.0234 | 2.3090 | 5.2926 | 2.3015 | 42.238 | 3.1096 |

**Table 3.8** *From rbox (200 random points, seed 34565473) and qhull (option v and o);* $d = 3$, $n^c = 71$.





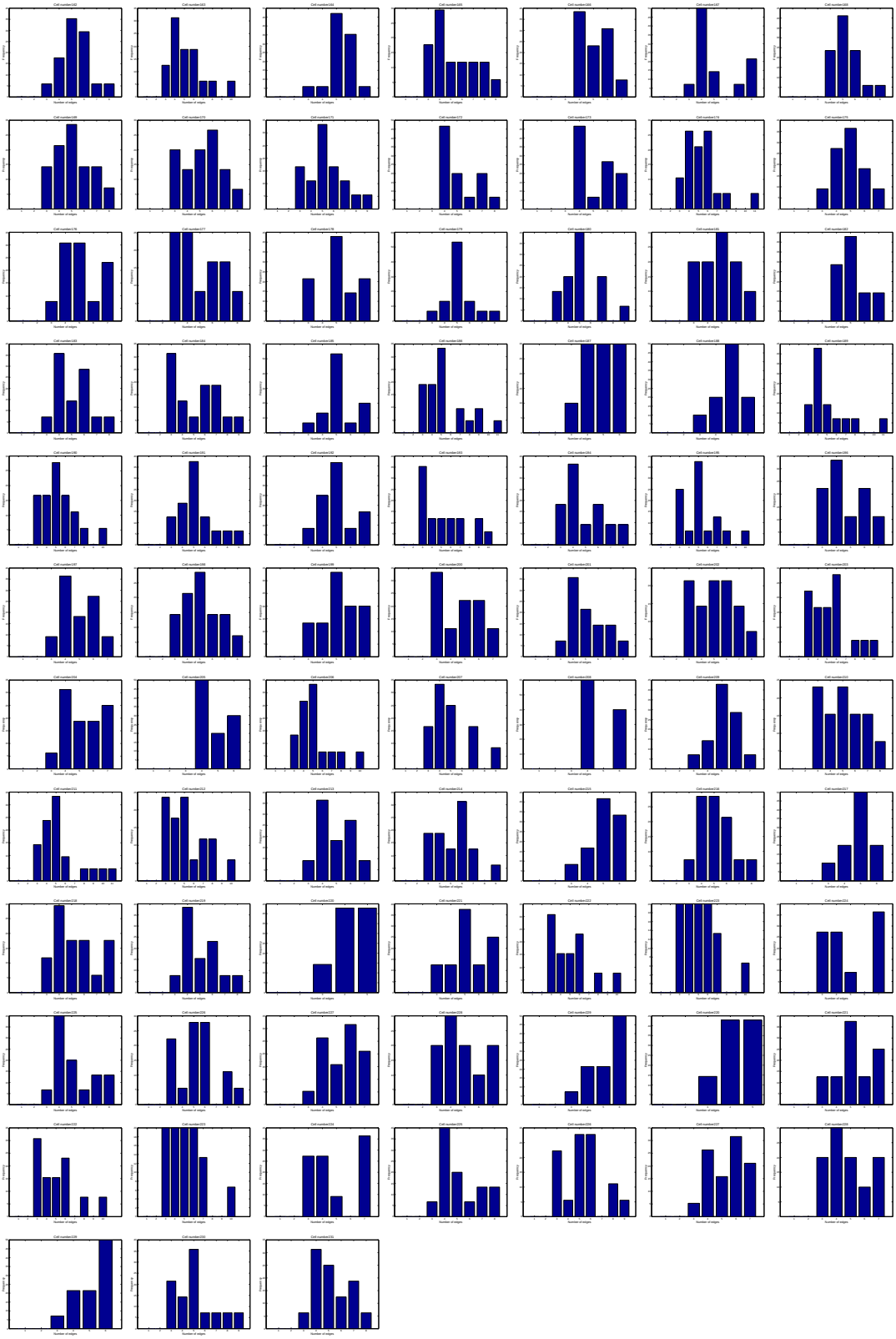


**Figure 3.27** *Distribution of the number of edges per face. Each picture is an individual cell. The distribution shows the relative abundance or the number of faces (the vertical axes) having the number of edges as shown by the horizontal axes. The horizontal axis scales are positive integers starting from zero at the origin. Simulation uses rbox (500 random points, seed 893280) and qhull (option v and o); $d = 3$, $n^c = 231$, $n^v = 3{,}107$, CPU time 81.1 sec for finding face area, 2.22 sec for cell volume, 49.59 sec for counting edges and 0.03 sec for finding the number of edges and faces.*

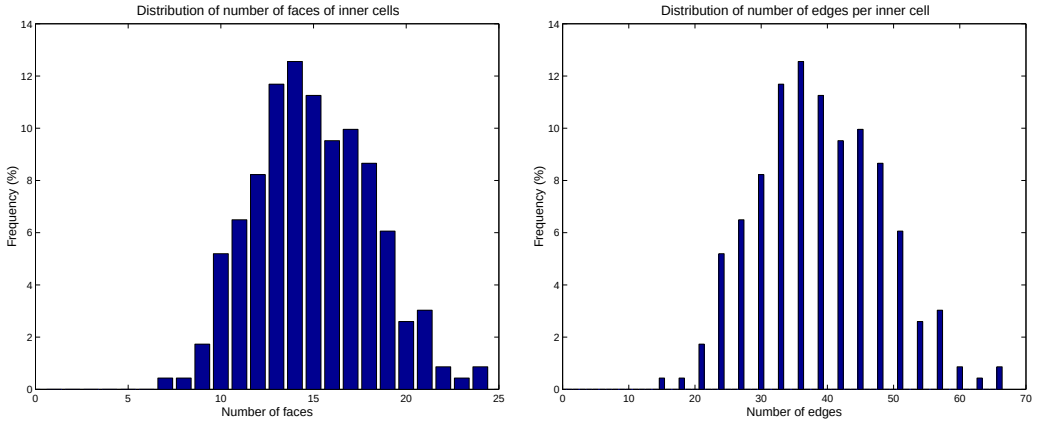
The minimum number of edges for each face is three. This number is the same as the number of vertices of that face. From these figures most of the cells have at least one face with three edges. There are only 19 cells (8.23 per cent) which does not have any three-edged face, and all of them have some four-edged faces. Therefore there is no cell with five as the minimum number of edges per face. The maximum number of edges per face is less clear-cut. There are twelve cells (5.19 per cent) with 11 as the maximum number of edges per face and two (0.87 per cent) with 12.

| | $n_c^v$ | $\aleph_c$ | $A_c^{fr}$ | $V_c^{fr}$ | $\alpha$ | $n_c^e$ | $E(n_{f,c}^e)$ | $\sigma(n_{f,c}^e)$ |
|---|---|---|---|---|---|---|---|---|
| min | 10 | 7 | $1.0989 \times 10^{-3}$ | $2.5790 \times 10^{-4}$ | 0.0406 | 15 | 4.2857 | 0.6030 |
| max | 44 | 24 | $1.0956 \times 10^{-2}$ | 0.24840 | 4.2505 | 66 | 5.5000 | 2.7028 |
| $\mu$ | 25.974 | 14.987 | $4.3290 \times 10^{-3}$ | $4.3290 \times 10^{-3}$ | 1.2515 | 38.961 | 5.1601 | 1.5450 |
| $\sigma^2$ | 40.852 | 10.213 | $3.3308 \times 10^{-6}$ | $2.9419 \times 10^{-4}$ | 0.3802 | 91.916 | 0.0372 | 0.1243 |
| $\sigma$ | 6.3915 | 3.1958 | $1.8250 \times 10^{-3}$ | $1.7152 \times 10^{-2}$ | 0.6166 | 9.5873 | 0.1928 | 0.3526 |
| $\mu_g$ | 25.166 | 14.642 | $3.9633 \times 10^{-3}$ | $1.8465 \times 10^{-3}$ | 1.0816 | 37.749 | 5.1564 | 1.5030 |
| $\mu_h$ | 24.323 | 14.288 | $3.6026 \times 10^{-3}$ | $1.2471 \times 10^{-3}$ | 0.8241 | 36.484 | 5.1526 | 1.4580 |
| med | 26 | 15 | $4.1467 \times 10^{-3}$ | $1.5715 \times 10^{-3}$ | 1.1915 | 39 | 5.2000 | 1.5315 |
| mad | 5.1532 | 2.5766 | $1.4064 \times 10^{-3}$ | $4.5864 \times 10^{-3}$ | 0.4720 | 7.7298 | 0.1505 | 0.2794 |
| $\mathcal{M}^2$ | 40.675 | 10.169 | $3.3163 \times 10^{-6}$ | $2.9291 \times 10^{-4}$ | 0.3785 | 91.518 | 0.0370 | 0.1238 |
| $\mathcal{M}^3$ | 59.377 | 7.4222 | $5.7258 \times 10^{-9}$ | $6.3817 \times 10^{-5}$ | 0.2143 | $2.0040 \times 10^2$ | -0.0071 | 0.0058 |
| $\mathcal{M}^4$ | $4.5398 \times 10^3$ | $2.8374 \times 10^2$ | $4.6684 \times 10^{-11}$ | $1.5395 \times 10^{-5}$ | 0.7390 | $2.2983 \times 10^4$ | 0.0063 | 0.0463 |
| $\mathcal{K}$ | 2.7440 | 2.7440 | 4.2448 | $1.7943 \times 10^2$ | 5.1578 | 2.7440 | 4.5730 | 3.0230 |

**Figure 3.28** *Statistics of a 231 cells Voronoi structure.*

Because the Voronoi tessellation being studied is simple, $\aleph_c^v = \aleph_c^f = n_c^f$. In other words, any two cells having at least one vertex in common are neighbours to each other, and the number of neighbours around any cell in an infinite network is equal to the number of its faces.

Figure 3.29 shows the distributions of the number of faces and edges per cell.


Distribution of number of faces of inner cells
Frequency (%)
Number of faces
Distribution of number of edges per inner cell
Frequency (%)
Number of edges

(a) (b)

**Figure 3.29** *The distributions of the number of* (a) *faces, and* (b) *edges, per cell.*

The results from 527 Voronoi cells are shown next. Table 3.9 is from a 3-d Voronoi structure generated from 1,000 cells. The creation codes started with a random seed of 234985. They spent 6,466.99 seconds for the counting of statistics, 270.56 for finding area of the faces, 4.47 for calculating cell volume and additional 2.8 seconds to find statistics for the number of edges. There are 6,357 vertices generated, plus another added distant vertice. Boundary cells were discarded, which leaves us with 527 inner cells which are taken into account.

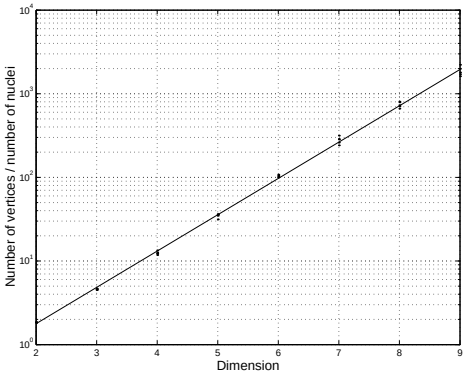
| | $v_{cin}$ | $\aleph_{v,cin}$ | $A_{fr,cin}$ | $V_{fr,cin}$ | $n_{f,cin}$ | $n_{e,cin}$ | |
|---|---|---|---|---|---|---|---|
| Min | 8 | 6 | $2.2191 \times 10^{-4}$ | $9.9722 \times 10^{-5}$ | 6 | 12 | |
| Max | 46 | 25 | $4.7015 \times 10^{-3}$ | 0.14364 | 25 | 69 | |
| $\mu$ | 26.338 | 15.169 | $1.8975 \times 10^{-3}$ | $1.8975 \times 10^{-3}$ | 15.169 | 39.507 | |
| $\sigma$[2] | 40.608 | 10.152 | $5.2144 \times 10^{-7}$ | $4.2313 \times 10^{-5}$ | 10.152 | 91.368 | |
| $\sigma$ | 6.3725 | 3.1862 | $7.2210 \times 10^{-4}$ | $6.5048 \times 10^{-3}$ | 3.1862 | 9.5587 | |
| $\mu_g$ | 25.543 | 14.829 | $1.7572 \times 10^{-3}$ | $1.1575 \times 10^{-3}$ | 14.829 | 38.315 | |
| $\mu_h$ | 24.703 | 14.478 | $1.6026 \times 10^{-3}$ | $8.5548 \times 10^{-4}$ | 14.478 | 37.054 | |
| Med | 26 | 15 | $1.8125 \times 10^{-3}$ | $1.1660 \times 10^{-3}$ | 15 | 39 | |
| $\delta_\mu$ | 5.0068 | 2.5034 | $5.6975 \times 10^{-4}$ | $1.4250 \times 10^{-3}$ | 2.5034 | 7.5103 | |
| $M$[2] | 40.531 | 10.133 | $5.2045 \times 10^{-7}$ | $4.2233 \times 10^{-5}$ | 10.133 | 91.195 | |
| $M$[3] | 72 | 9 | $2.5644 \times 10^{-10}$ | $5.4435 \times 10^{-6}$ | 9 | 243 | |
| $M$[4] | 5088.9 | 318.06 | | $1.0467 \times 10^{-12}$ | $7.6653 \times 10^{-7}$ | 318.06 | 25763 |
| $\kappa$ | 3.0978 | 3.0978 | 3.8644 | 429.77 | 3.0978 | 3.0978 | |

**Table 3.9** *Statistics from 527 Voronoi cells.*

The mean value of the mean numbers of edges per face obtained was 5.1712. This can be called $\mathrm{E}(\mathrm{E}(n_{e,f}))$, the expected value over all cells of the mean number of edges in a face averaged over all its faces. The variance was 0.035983, minimum value 4, maximum 5.52, standard deviation 0.18969, geometric mean 5.1676, harmonic mean 5.1638, median 5.2, mean absolute deviation 0.14465, $2^{nd}$ moment 0.035915, $3^{rd}$ moment -0.0083404, and kurtosis 6.1689. All the mean numbers for each face are 4.9091, 5.0769, 5.2500, 5.1429, 4.8000, 4.9091, 5.2500, 5.1429, 5.1429, 5.0769, 5.2000, 5.2000, 5.0000, 5.2500, 4.8000, 5.2941, 5.0769, 5.2000, 4.9091, 5.0769, 4.6667, 5.3684, 5.2000, 5.3333, 4.6667, 5.2500, 5.0000, 4.5000, 5.0000, 5.4286, 5.2000, 5.2500, 5.0769, 5.2500, 5.1429, 4.9091, 5.0000, 5.0000, 5.3333, 4.8000, 5.3333, 5.2000, 5.3684, 5.2500, 5.0000, 5.0769, 5.0769, 5.2941, 4.8000, 5.0000, 5.0769, 5.4286, 5.3333, 5.1429, 4.6667, 4.6667, 5.4783, 5.2941, 5.2500, 5.2500, 4.9091, 4.9091, 5.4000, 5.2500, 5.1429, 5.2941, 5.4545, 5.1429, 5.0000, 4.9091, 5.2500, 5.1429, 5.1429, 5.4545, 4.8000, 5.1429, 5.0000, 5.2941, 5.2941, 5.0000, 5.4286, 5.0769, 5.1429, 5.1429, 5.2941, 5.0769, 4.9091, 5.2000, 5.2000, 5.4545, 5.1429, 5.2500, 5.1429, 5.2000, 5.2500, 5.1429, 5.1429, 5.0000, 5.3684, 5.3684, 5.2941, 5.4000, 5.4783, 5.3684, 4.9091, 5.2941, 5.2500, 4.8000, 5.2500, 5.2941, 5.1429, 5.2500, 5.2500, 5.2000, 5.0769, 5.4000, 5.0000, 5.3333, 5.4286, 4.8000, 5.2500, 5.3333, 5.0769, 4.9091, 5.3333, 5.3333, 5.2941, 5.2000, 5.3333, 5.3333, 5.2500, 5.2500, 5.2500, 5.2000, 5.2000, 5.1429, 5.0000, 4.8000, 5.3684, 5.1429, 5.2000, 5.2500, 5.0769, 5.3684, 5.2941, 5.1429, 5.2500, 5.2000, 5.1429, 4.8000, 5.0000, 5.3333, 5.2000, 5.0769, 5.2000, 5.3333, 4.5000, 5.0000, 5.3333, 5.3333, 5.0000, 5.2500, 5.2000, 4.9091, 5.1429, 5.0000, 5.3684, 5.2500, 5.0000, 5.4000, 5.3333, 5.2500, 4.8000, 5.5000, 5.1429, 5.2000, 5.2500, 5.2941, 5.2500, 5.2941, 5.4000, 5.2941, 5.2941, 4.9091, 5.2000, 5.2000, 5.4286, 5.0769, 5.3684, 5.2500, 5.3333, 5.0000, 4.9091, 5.3684, 4.0000, 5.1429, 5.3333, 4.9091, 5.3333, 5.2000, 5.1429, 5.1429, 5.1429, 5.0000, 5.0769, 5.1429, 5.3684, 4.9091, 5.2941, 5.0000, 5.2941, 5.2941, 5.2500, 5.1429, 5.0769, 5.2000, 5.4000, 5.1429, 5.0769, 5.2000, 4.6667, 5.3684, 5.4286, 5.0769, 5.2500, 5.1429, 5.3684, 5.0769, 5.2500, 5.3684, 5.3684, 5.0769, 4.8000, 5.3333, 5.0769, 5.1429, 5.2000, 4.9091, 5.2000, 4.9091, 5.2500, 5.2000, 4.9091, 5.2500, 5.2941, 5.5000, 5.2000, 5.2500, 5.4783, 5.2941, 5.2000, 5.0000, 5.2000, 5.2941, 5.2000, 5.2941, 4.9091, 5.4286, 5.2500, 5.1429, 5.5000, 5.1429, 5.4286, 5.0000, 5.3684, 5.2500, 5.3684, 5.4000, 5.2500, 5.2941, 4.8000, 4.9091, 4.8000, 5.1429, 5.3333, 5.2000, 5.0000, 5.0769, 5.0000, 5.0769, 5.1429, 5.1429, 5.0769, 5.0769, 5.4545, 5.0769, 5.2000, 5.0769, 5.4545, 5.4286, 4.8000, 5.2000, 5.0769, 5.3684, 5.3333, 5.3333, 5.4000, 5.2000, 5.0000, 5.0000, 5.2941, 5.2000, 4.9091, 5.2500, 5.3333, 5.0000, 5.3684, 5.3333, 5.3333, 5.0000, 5.4545, 5.1429, 5.0769, 5.0769, 5.0000, 5.5200, 5.0769, 5.1429, 5.2000, 5.3333, 5.0000, 5.2941, 5.2500, 5.2500, 5.2500, 5.2000, 5.2000, 5.2000, 5.2000, 5.2000, 5.2000, 5.2000, 5.2500, 5.2500, 4.9091, 5.4000, 5.2941, 5.2000, 5.2500, 5.2500, 5.1429, 5.0769, 5.0769, 5.2000, 5.2941, 5.2941, 5.4783, 5.3333, 4.8000, 5.3333, 5.2941, 5.0769, 5.3333, 5.1429, 5.3684, 5.2941, 5.1429, 5.1429, 5.4545, 5.3333, 4.9091, 5.1429, 5.2941, 5.2500, 5.0769, 5.3333, 4.9091, 5.0769, 5.3333, 5.0000, 4.9091, 5.3333, 5.2941, 5.3684, 5.2941, 5.1429, 5.2500, 5.3333, 5.1429, 5.0769, 5.2000, 5.2500, 5.3333, 5.0769, 5.0000, 5.2941, 4.9091, 5.0769, 5.0769, 5.4545, 5.2000, 5.2941, 4.8000, 5.2941, 5.2500, 5.3684, 5.2000, 4.9091, 5.2000, 5.2000, 5.1429, 4.6667, 5.2000, 5.2500, 4.6667, 5.3333, 5.2941, 5.0769, 5.0769, 5.0000, 5.3333, 5.0000, 5.4000, 5.4545, 5.3333, 5.2500, 5.1429, 5.2000, 4.9091, 5.1429, 5.2000, 5.3333, 5.2941, 5.5000, 5.2500, 5.1429, 5.2000, 5.3684, 5.1429, 5.4286, 5.0769, 5.2941, 5.2941, 5.2500, 5.1429, 5.1429, 5.2500, 5.1429, 5.3333, 4.8000, 5.2000, 5.2000, 4.8000, 5.4286, 5.2000, 5.2941, 5.2500, 5.5000, 5.2941, 5.0000, 5.2941, 5.2500, 4.6667, 5.2941, 5.0000, 5.2500, 5.2500, 4.9091, 5.2000, 5.2941, 5.0769, 5.3684, 4.9091, 4.9091, 5.2000, 5.4000, 5.2000, 5.0769, 5.3684, 5.2000, 5.2500, 5.0769, 5.2500, 5.0000, 5.2941, 4.5000, 5.3333, 5.2000, 5.2941, 5.1429, 5.0000, 5.0769, 4.5000, 5.4783, 5.3684, 5.1429, 4.9091, 5.3684, 5.2000, 5.2500, 5.0000, 5.2941, 5.0000,

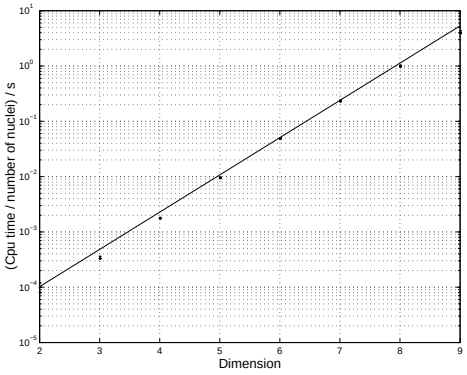
5.0769, 5.0000, 5.3333, 5.2000, 5.2000, 5.2500, 5.2000, 5.2000, 5.2000, 4.8000, 5.0769, 5.2000, 5.2941, 5.2941, 5.2500, 5.0000, 5.4286, 5.2000, 5.2500, 5.3333, 5.2000, 5.2000, 5.0000, 5.1429, 5.2000, 5.2941, 5.2941, 5.3333, 4.8000, 5.3333, 4.6667, 5.1429, 5.1429 and 5.2000.

Then there are the standard deviations. The standard deviation of $\mathrm{E}(n_{e,f})$ averaged over all faces of bounded cells is 1.5586. The rest are $\sigma^2 = 0.1186$, $\sigma = 0.3444$, minimum 0.6030, maximum 2.5690, $\mu_g = 1.5169$, $\mu_h = 1.4700$, median 1.5706, $\delta_\mu = 0.2715$, $M^2 = 0.1184$, $M^3 = -0.0060$, and $\kappa = 2.9916$. All these standard deviations are 1.4460, 1.5525, 1.5706, 2.3157, 1.2293, 1.5783, 1.5275, 1.0995, 1.0271, 1.7541, 1.4736, 1.2649, 1.5954, 1.2910, 0.6325, 1.5315, 0.9541, 1.2071, 0.7006, 0.8623, 1.0000, 1.3829, 1.5675, 1.7150, 1.2247, 1.6533, 1.5374, 0.9258, 1.1282, 1.8323, 1.9346, 1.3416, 0.6405, 1.8439, 1.6575, 1.3003, 1.4142, 1.5374, 1.5339, 1.1353, 1.5718, 0.8619, 2.1137, 1.7321, 1.5954, 1.6053, 1.3205, 1.7235, 0.7888, 1.2792, 1.7059, 1.6301, 1.4142, 1.6575, 1.3229, 0.8660, 1.9038, 1.9610, 1.9149, 1.8074, 0.9439, 1.5136, 1.9841, 1.3904, 1.6575, 1.2632, 2.0172, 1.7913, 1.5954, 1.9212, 1.4376, 1.9945, 1.6104, 1.5954, 1.1353, 1.8752, 1.4142, 1.7235, 1.4902, 1.4142, 1.5991, 1.3821, 1.8337, 1.0271, 1.6494, 1.0377, 2.0226, 1.6987, 1.9712, 1.2994, 1.5119, 0.9309, 1.9555, 1.0142, 1.0646, 1.2924, 2.1070, 1.8586, 2.0873, 1.4225, 2.2573, 1.7290, 1.7547, 1.9779, 1.6404, 1.7946, 2.2361, 1.4757, 1.6125, 1.4038, 0.9493, 1.6533, 1.5706, 1.6125, 1.8467, 1.4290, 1.0445, 1.7489, 1.5675, 1.3984, 1.1832, 1.9704, 0.9541, 1.3003, 2.1420, 1.6803, 1.6111, 2.2104, 1.9704, 1.5718, 1.2383, 1.3416, 1.6125, 2.1778, 1.5675, 1.6104, 1.1282, 1.3166, 2.0605, 1.9945, 1.2649, 1.4832, 1.3821, 2.0873, 1.4902, 1.4601, 1.2910, 1.5213, 1.4601, 0.6325, 1.8586, 2.5437, 1.7809, 1.3821, 1.5675, 2.1420, 1.0690, 1.0445, 1.6803, 1.8787, 1.2792, 1.4832, 1.4736, 1.2210, 1.6104, 0.6030, 1.6059, 1.5706, 1.4771, 1.6026, 1.6803, 1.3416, 1.3166, 2.0430, 1.1673, 1.6125, 1.6125, 1.6111, 1.9833, 1.6494, 1.8180, 1.6111, 1.4476, 0.9439, 1.6987, 1.5675, 1.6301, 1.3821, 1.8016, 2.1448, 1.0290, 1.7056, 1.4460, 2.1137, 0.8944, 1.8752, 1.9704, 1.4460, 1.1376, 1.5213, 1.8337, 1.4601, 1.6575, 1.4142, 1.5525, 1.7478, 2.2413, 1.1362, 1.3585, 1.5374, 1.6111, 2.0238, 1.3904, 1.4601, 1.0377, 1.7809, 1.5009, 2.1432, 1.7541, 2.3964, 1.0000, 1.4985, 1.9640, 2.0191, 1.7701, 2.4450, 1.3829, 0.8623, 1.9833, 1.8918, 1.8622, 1.4412, 1.7512, 1.3284, 1.5525, 1.2924, 1.7809, 1.1362, 1.2649, 1.3003, 1.6931, 1.6562, 1.0445, 1.4376, 1.5315, 1.8415, 2.1112, 1.7701, 1.9038, 1.9289, 1.1464, 1.2060, 1.3202, 1.6494, 1.2071, 1.7235, 1.8141, 2.2265, 1.7701, 1.4064, 2.0000, 1.4064, 1.7768, 1.0445, 1.8321, 2.1134, 1.9779, 1.6026, 1.5706, 1.5718, 1.5492, 0.9439, 1.5492, 0.9493, 1.4552, 1.8593, 0.9535, 1.1875, 1.9069, 1.3205, 1.4601, 1.5119, 1.1875, 1.7059, 1.7107, 1.4979, 1.6987, 1.1152, 1.9451, 1.8860, 1.3984, 1.3732, 1.4979, 1.7388, 1.7150, 1.9403, 1.1877, 1.7809, 1.0445, 1.2060, 1.3585, 1.4243, 1.3003, 1.1832, 1.5339, 0.9535, 2.2659, 1.1882, 1.6450, 1.4142, 1.7655, 0.9493, 1.3205, 1.7059, 1.7581, 1.2949, 1.0377, 1.8337, 1.3732, 1.4951, 1.5374, 1.4038, 1.6931, 1.6125, 1.4832, 1.8974, 1.8205, 1.5675, 1.3202, 1.4736, 1.9712, 1.8974, 1.9149, 1.5275, 1.8141, 1.6026, 1.5315, 1.7809, 2.5690, 1.8074, 1.7033, 1.6053, 0.9541, 1.3202, 1.7235, 1.7594, 1.2746, 1.1882, 1.3984, 1.4142, 1.6111, 1.3205, 1.9097, 1.7913, 1.8016, 1.3117, 1.4601, 1.6575, 1.6250, 2.0000, 1.6404, 1.6104, 1.4476, 1.5706, 1.3205, 1.3284, 1.1362, 1.8467, 1.6088, 1.8091, 1.5783, 1.4552, 2.2013, 1.6401, 1.6494, 1.8752, 1.9494, 1.6450, 1.7033, 1.7059, 2.1112, 1.8074, 1.5718, 1.1875, 1.1282, 1.3585, 1.1362, 1.9774, 1.1875, 2.0639, 1.2649, 1.1600, 1.1353, 1.8962, 1.5706, 1.8321, 1.8974, 1.8141, 1.3202, 1.4736, 1.7033, 1.4142, 1.5675, 1.5275, 0.7071, 2.2229, 1.4038, 1.2558, 1.5525, 1.5954, 1.3284, 2.0449, 2.2572, 1.8702, 1.6088, 1.4376, 1.0995, 1.7809, 0.8312, 1.7478, 1.6125, 1.3720, 2.0544, 1.8178, 2.0494, 1.6575, 1.9712, 2.0873, 1.7033, 1.7768, 1.7541, 1.9610, 1.2632, 1.4832, 1.4601, 1.3506, 1.7321, 1.5619, 1.9097, 1.5492, 1.7809, 1.3202, 1.2293, 1.5991, 1.7403, 1.9289, 1.3904, 1.7937, 1.4476, 1.0445, 1.6494, 1.6533, 1.3229, 1.7594, 1.2792, 1.6931, 1.7321, 0.8312, 1.0142, 1.8962, 1.6564, 1.5352, 1.5136, 1.0445, 1.6987, 1.5694, 1.6987, 1.7059, 1.7065, 1.7403, 1.2383, 1.5525, 1.8074, 1.2792, 2.1437, 1.0690, 1.5718, 1.6562, 2.0544, 1.2924, 1.8091, 2.3616, 1.5119, 1.4419, 1.4985, 1.1673, 1.2210, 2.1657, 1.7403, 2.0166, 1.4142, 1.9610, 1.3484, 1.0377, 1.4771, 2.1693, 1.4243, 1.5213, 1.2383, 2.0771, 1.7809, 1.7809, 0.6325, 1.5525, 1.3202, 0.9196, 1.6111, 2.1448, 1.7581, 1.7485, 1.6125, 1.8074, 1.9704, 1.7809, 0.8619, 1.2792, 1.8337, 1.7809, 1.6494, 1.7235, 1.3720, 0.7888, 1.7489, 0.8660, 1.3506, 1.5119 and 1.5213.




The line shown in Figure 3.31 is $y = 0.2410e^n$, where $n$ is the dimension of the network and is the horizontal axis; the coefficient of the exponential term is obtained by averaging over the averages in each dimension, each of which in turn comes from five batch runs.

**Figure 3.31** Ratios between the number of vertices and the number of cells in Voronoi networks of various dimensions..




The line shown in Figure 3.33, found manually by trial and error, has the equation $y = 4.61 \times 10^{-6}(2 + e)^n$. In comparison, substituting the average cpu time for each dimension for $y$ in the equation $y = Ae^n$ to obtain $A$ and then find the average again over all dimensions results in $\bar{A} = 1.612 \times 10^{-4}$. The program which produces both Figure 3.33 and ℵ is given in § A.9.

**Figure 3.33** Cpu time in creating the Voronoi networks for Figure 3.33..

The number of vertices of higher dimensions is investigated briefly in the following Table 3.10 and Figure 3.34 the simulation of which was carried out by the program in § A.16.

| **Dimension** | 4 | 5 | 6 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|
| $N_c$ | 300 | 300 | 300 | 30 | 30 | 30 |
| $N_v$ | 6,577 | 27,150 | 118,534 | 5,465 | 10,467 | 17,442 |
| $n_c$ | 132 | 104 | 14 | 2 | 3 | 0 |
| $\min(n_c^v)$ | 26 | 244.00 | 1,020.0 | 952.002 | 2,353.0 | 4,287.0 |
| $\max(n_c^v)$ | 255 | 1,142.0 | 5,852.0 | 2,638.0 | 5,534.0 | 9,528.0 |
| $\bar{n}_c^v$ | 109.90 | 543.51 | 2,766.5 | 1,640.2 | 3,489.7 | 6,396.0 |
| $(\sigma^2_{n_v})_c$ | 1,155.9 | 22,974 | $6.0898 \times 10^5$ | $1.7964 \times 10^5$ | $7.0328 \times 10^5$ | $2.3635 \times 10^6$ |
| $\sigma_c^{n_v}$ | 33.999 | 151.57 | 780.37 | 423.84 | 838.62 | $1.5374 \times 10^3$ |
| $\mu_g(n_c^v)$ | 104.41 | 523.06 | 2,666.1 | 1,589.9 | 3,398.4 | 6,225.6 |
| $\mu_h(n_c^v)$ | 98.389 | 502.94 | 2,571.4 | 1,542.3 | 3,313.9 | 6,065.2 |
| $\text{med}(n_c^v)$ | 107.50 | 527.00 | 2,651.0 | 1,568.5 | 3,272.5 | 5,816.5 |
| $\text{mad}(n_c^v)$ | 27.366 | 119.39 | 588.19 | 331.96 | 687.47 | 1,296.6 |
| $m^2(n_c^v)$ | 1,152.1 | 22,897 | $6.0695 \times 10^5$ | $1.7365 \times 10^5$ | $6.7984 \times 10^5$ | $2.2847 \times 10^6$ |
| $m^3(n_c^v)$ | 15,555 | $2.2871 \times 10^6$ | $5.1556 \times 10^8$ | $4.7965 \times 10^7$ | $3.9108 \times 10^8$ | $1.8988 \times 10^9$ |
| $m^4(n_c^v)$ | $4.5760 \times 10^6$ | $1.8905 \times 10^9$ | $1.7964 \times 10^{12}$ | $8.5043 \times 10^{10}$ | $1.1967 \times 10^{12}$ | $1.1638 \times 10^{13}$ |
| $\kappa(n_c^v)$ | 3.4476 | 3.6059 | 4.8764 | 2.8202 | 2.5892 | 2.2296 |

**Table 3.10** *Statistics of the number of vertices in 4, 5, 6, 8, 9, and 10 dimensions*

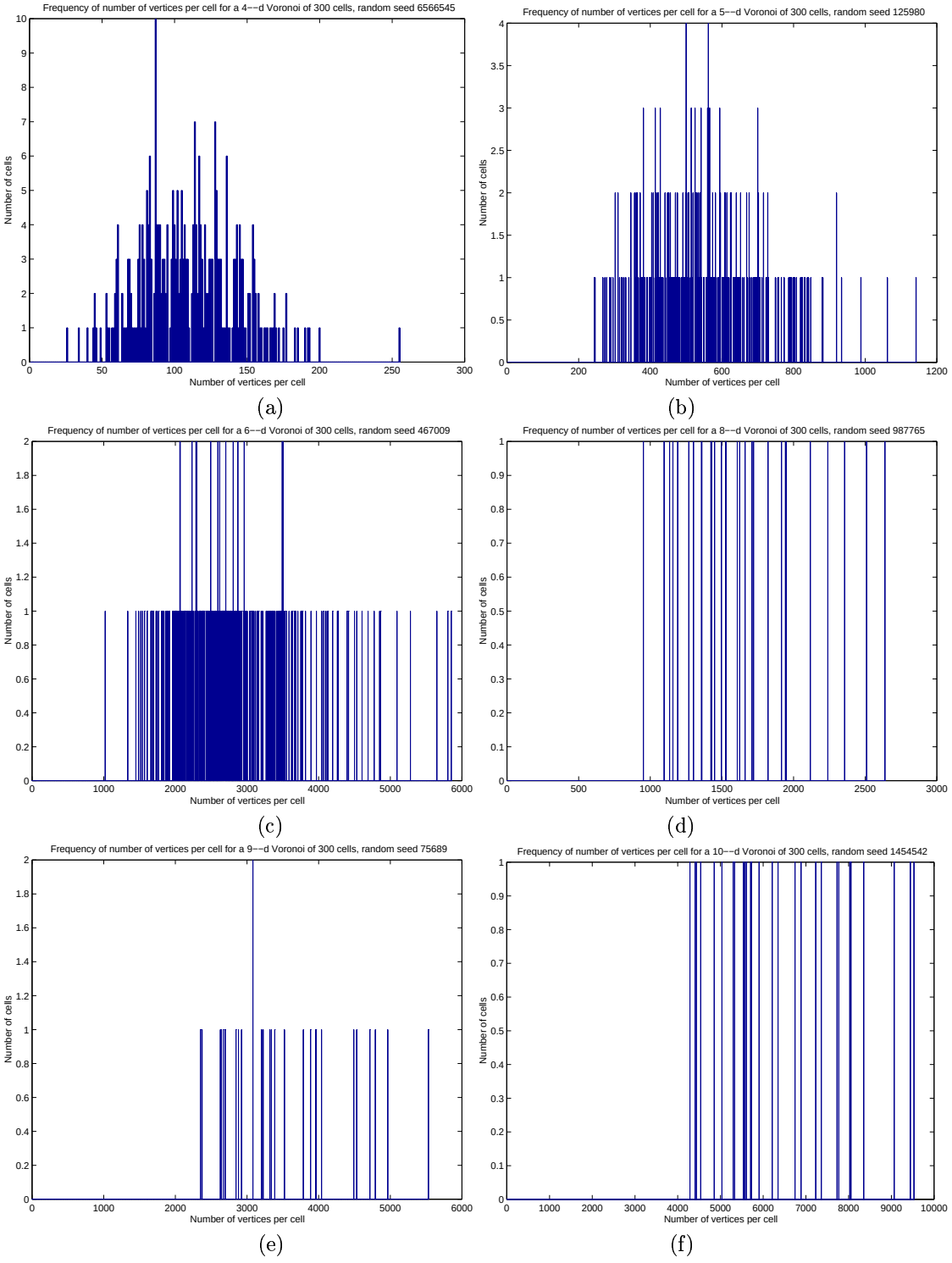
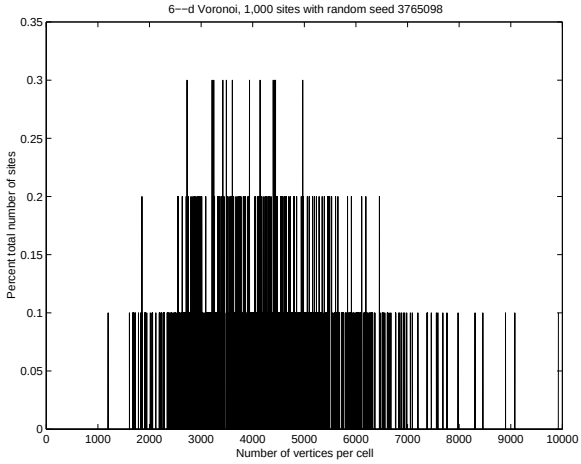

Frequency of number of vertices per cell for a 4−−d Voronoi of 300 cells, random seed 6566545
Number of cells
Number of vertices per cell
(a)
Frequency of number of vertices per cell for a 5−−d Voronoi of 300 cells, random seed 125980
Number of cells
Number of vertices per cell
(b)
Frequency of number of vertices per cell for a 6−−d Voronoi of 300 cells, random seed 467009
Number of cells
Number of vertices per cell
(c)
Frequency of number of vertices per cell for a 8−−d Voronoi of 300 cells, random seed 987765
Number of cells
Number of vertices per cell
(d)
Frequency of number of vertices per cell for a 9−−d Voronoi of 300 cells, random seed 75689
Number of cells
Number of vertices per cell
(e)
Frequency of number of vertices per cell for a 10−−d Voronoi of 300 cells, random seed 1454542
Number of cells
Number of vertices per cell
(f)

**Figure 3.34** *Distribution of vertices in* (a) *4,* (b) *5,* (c) *6, and* (d) *8,* (e) *9,* (f) *10 dimensions*

To obtain the number of vertices per cell, $n_c^v$, of a six-dimensional Voronoi structure of 1,000 cells I used a batch program, for instance the one listed in § A.17. The Matlab macro that this program refers to opens and reads from a file the number of vertices and then finds the statistical values, *viz.* $\min(n_c^v) = 1{,}198$; $\max(n_c^v) = 9{,}923$; $\bar{n}_c^v = 4{,}201.1$; $(\sigma^2_{n_v})_c = 1.4069 \times 10^6$; $\sigma_c^{n_v} = 1186.1$; $(\bar{n}_g)_c^v = 4035.4$; $(\bar{n}_h)_c^v = 3866.8$; $\mathrm{med}(n_c^v) = 4122.5$; $\mathrm{mad}(n_c^v) = 931.81$; $m^2(n_c^v) = 1.4055 \times 10^6$; $m^3(n_c^v) = 1.0462 \times 10^9$; $m^4(n_c^v) = 7.7148 \times 10^{12}$; and $\kappa(n_c^v) = 3.9053$.

The following figure shows the frequency of the number of vertices.




There are three cells with 2729, 3213, 3246, 3421, 3490, 3606, 3938, 4143, 4398, 4417, 4442 and 4970 vertices. There are two having 1854, 2549, 2557, 2634, 2712, 2720, 2722, 2751, 2804, 2843, 2869, 2878, 2882, 2920, 2931, 2957, 2973, 2996, 3013, 3090, 3260, 3322, 3329, 3343, 3366, 3393, 3418, 3424, 3446, 3513, 3520, 3560, 3577, 3583, 3585, 3610, 3631, 3677, 3714, 3732, 3753, 3767, 3770, 3819, 3835, 3861, 3907, 3919, 3924, 3937, 3939, 4045, 4053, 4091, 4117, 4122, 4123, 4163, 4178, 4219, 4251, 4261, 4269, 4272, 4298, 4317, 4327, 4355, 4461, 4470, 4473, 4474, 4542, 4563, 4572, 4576, 4586, 4618, 4624, 4629, 4640, 4646, 4648, 4700, 4729, 4792, 4810, 4851, 4942, 4975, 4984, 4985, 5058, 5064, 5097, 5161, 5195, 5235, 5287, 5347, 5387, 5455, 5467, 5492, 5529, 5606, 5650, 5838, 5913, 6112, 6193 and 6455 vertices. And there is only one cell with each of the following numbers of vertices, 1198, 1612, 1672, 1686, 1688, 1717, 1727, 1787, 1827, 1864, 1906, 1915, 1921, 1947, 2016, 2052, 2056, 2060, 2123, 2190, 2200, 2223, 2231, 2236, 2267, 2280, 2281, 2286, 2289, 2344, 2353, 2356, 2371, 2372, 2385, 2408, 2423, 2426, 2429, 2455, 2475, 2487, 2499, 2500, 2503, 2504, 2508, 2513, 2542, 2547, 2551, 2560, 2561, 2565, 2575, 2578, 2580, 2585, 2586, 2596, 2617, 2619, 2622, 2650, 2654, 2658, 2664, 2669, 2673, 2686, 2692, 2694, 2704, 2708, 2716, 2718, 2723, 2732, 2733, 2742, 2743, 2755, 2771, 2779, 2781, 2783, 2791, 2800, 2807, 2808, 2811, 2820, 2835, 2837, 2853, 2858, 2864, 2867, 2870, 2873, 2875, 2880, 2884, 2887, 2893, 2894, 2895, 2902, 2907, 2910, 2914, 2916, 2925, 2928, 2934, 2949, 2955, 2956, 2960, 2967, 2974, 2978, 2983, 2985, 2989, 2993, 3007, 3012, 3014, 3027, 3051, 3074, 3102, 3107, 3114, 3116, 3119, 3129, 3130, 3136, 3137, 3141, 3148, 3152, 3164, 3173, 3176, 3180, 3182, 3186, 3188, 3190, 3192, 3199, 3202, 3204, 3206, 3208, 3219, 3221, 3229, 3231, 3237, 3239, 3244, 3245, 3255, 3256, 3259, 3267, 3270, 3271, 3274, 3275, 3285, 3295, 3296, 3305, 3307, 3312, 3316, 3317, 3318, 3319, 3333, 3337, 3344, 3351, 3352, 3354, 3357, 3360, 3370, 3373, 3374, 3378, 3384, 3390, 3394, 3396, 3402, 3403, 3408, 3427, 3428, 3436, 3440, 3443, 3444, 3452, 3453, 3454, 3486, 3487, 3492, 3499, 3502, 3505, 3506, 3509, 3519, 3521, 3522, 3523, 3536, 3541, 3544, 3545, 3547, 3549, 3551, 3555, 3556, 3573, 3575, 3578, 3581, 3582, 3587, 3589, 3594, 3595, 3599, 3604, 3609, 3611, 3620, 3628, 3634, 3635, 3636, 3639, 3647, 3652, 3656, 3658, 3663, 3666, 3668, 3680, 3685, 3686, 3687, 3689, 3691, 3694, 3695, 3696, 3706, 3709, 3710, 3713, 3716, 3717, 3727, 3728, 3735, 3739, 3740, 3742, 3743, 3744, 3751, 3764, 3772, 3775, 3776, 3778, 3782, 3787, 3789, 3792, 3794, 3798, 3803, 3804, 3806, 3808, 3810, 3815, 3818, 3821, 3830, 3834, 3837, 3838, 3842, 3847, 3849, 3850, 3863, 3867, 3868, 3883, 3884, 3889, 3890, 3899, 3900, 3902, 3903, 3908, 3912, 3914, 3918, 3920, 3925, 3929, 3942, 3943, 3944, 3947, 3949, 3956, 3957, 3960, 3962, 3963, 3978, 3980, 3981, 3986, 3988, 3996, 4008, 4016, 4017, 4019, 4027, 4030, 4033, 4035, 4038, 4039, 4044, 4049, 4057, 4072, 4073, 4075, 4077, 4082, 4086, 4092, 4106, 4111, 4112, 4124, 4126, 4128, 4134, 4140, 4145, 4146, 4147, 4148, 4150, 4153, 4157, 4162, 4176, 4177, 4179, 4188, 4189, 4190, 4193, 4195, 4203, 4206, 4212, 4214, 4217, 4218, 4225, 4227, 4232, 4238, 4239, 4244, 4245, 4246, 4247, 4257, 4262, 4271, 4273, 4279, 4283, 4288, 4291, 4293, 4300, 4308, 4313, 4316, 4320, 4322, 4324, 4326, 4331, 4334, 4335, 4337, 4339, 4342, 4346, 4349, 4350, 4351, 4352, 4354, 4357, 4361, 4376, 4377, 4386, 4388, 4395, 4399, 4405, 4410, 4411, 4413, 4425, 4426, 4428, 4447, 4449, 4451, 4453, 4459, 4471, 4475, 4480, 4488, 4493, 4496, 4501, 4504, 4506, 4512, 4515, 4518, 4522, 4523, 4532, 4548, 4552, 4556, 4561, 4574, 4580, 4585, 4587, 4595, 4598, 4602, 4605, 4610, 4614, 4619, 4621, 4626, 4628, 4632, 4634, 4639, 4649, 4651, 4657, 4663, 4665, 4674, 4681, 4684, 4697, 4702, 4703, 4705, 4710, 4711, 4714, 4725, 4728, 4736, 4738, 4740, 4751, 4754, 4755, 4761, 4765, 4770, 4779, 4781, 4814, 4816, 4826, 4829, 4831, 4844, 4850, 4873, 4877, 4880, 4885, 4890, 4892, 4896, 4899, 4900, 4902, 4903, 4905, 4908, 4911, 4914, 4915, 4919, 4924, 4933, 4937, 4944, 4955, 4960, 4967, 4969, 4974, 4992, 4994, 4997, 5000, 5002, 5003, 5007, 5019, 5021, 5028, 5029, 5035, 5039, 5050, 5074, 5080, 5083, 5092, 5093, 5104, 5108, 5110, 5117, 5119, 5120, 5131, 5138, 5156, 5158, 5166, 5167,

5176, 5185, 5192, 5194, 5200, 5203, 5219, 5225, 5226, 5233, 5236, 5239, 5251, 5259, 5265, 5270, 5271, 5274, 5285, 5288, 5290, 5313, 5316, 5318, 5325, 5328, 5336, 5337, 5346, 5364, 5389, 5405, 5408, 5415, 5432, 5439, 5440, 5448, 5457, 5460, 5463, 5464, 5466, 5468, 5470, 5474, 5482, 5484, 5530, 5549, 5567, 5589, 5591, 5598, 5612, 5627, 5632, 5664, 5676, 5680, 5718, 5719, 5720, 5724, 5738, 5744, 5747, 5748, 5749, 5752, 5764, 5802, 5806, 5824, 5841, 5859, 5880, 5885, 5892, 5896, 5899, 5927, 5928, 5947, 5956, 5957, 5966, 5998, 6006, 6014, 6017, 6031, 6035, 6038, 6040, 6054, 6055, 6075, 6081, 6084, 6089, 6092, 6113, 6145, 6166, 6169, 6180, 6192, 6195, 6221, 6237, 6265, 6272, 6273, 6286, 6303, 6305, 6310, 6313, 6320, 6356, 6367, 6456, 6469, 6510, 6515, 6520, 6521, 6524, 6541, 6561, 6609, 6615, 6642, 6677, 6767, 6771, 6831, 6852, 6883, 6918, 6945, 6985, 7056, 7093, 7201, 7379, 7387, 7461, 7560, 7588, 7685, 7764, 7765, 7979, 8308, 8460, 8899, 9079 and 9923. But these results are not very useful since border cells are present.

**Result on 700 cells**

```
Box size: 10
No compression
Number of cells: 700
Number of vertices: 4380
Number of cells in frame: 341
Time for counting stats: 2191.39 seconds
Number of faces connected to the first vertice at infinity: 181
Time for finding cell volumes: 3.1500 seconds
```

| $n_c$ | $n_v$ | $n_{f,1^{st}v}$ | $n_{c_{in}}$ | $t_{CPU,stat.}$ | $t_{CPU,A}$ | $t_{CPU,V}$ | |
|---|---|---|---|---|---|---|---|
| 700 | 4,380 | 181 | 341 | 2191.39 sec. | 137.13 sec. | 3.15 sec. | |
| $\min n_{v_c}$ | $\max n_{v_c}$ | $\bar{n_{v_c}}$ | $\sigma^2_{n_{v_c}}$ | $\sigma_{n_{v_c}}$ | $g, \bar{n}_{v_c}$ | $h, \bar{n}_{v_c}$ | $n_{v_c}$ |
| 10 | 46 | 25.106 | 46.784 | 6.8399 | 24.169 | 23.208 | |
| $\widetilde{n_{v_c}}$ | $\delta_\mu(n_{v_c})$ | $M^2(n_{v_c})$ | $M^3(n_{v_c})$ | $M^4(n_{v_c})$ | $\kappa_{n_{v,c}}$ | | |
| 24 | 5.4023 | 46.717 | $1.4240\times10^2$ | $6.5877\times10^3$ | 3.0184 | | |
| $\min n_{v,c_{in}}$ | $\max n_{v,c_{in}}$ | $\bar{v_{c_{in}}}$ | $\sigma^2_{v,c_{in}}$ | $\sigma_{v,c_{in}}$ | $\bar{V}_{,gc_{in}}$ | $\bar{V}_{,hc_{in}}$ | $n_{v,c_{in}}$ |
| 10 | 46 | 26.246 | 40.974 | 6.4011 | 25.464 | 24.668 | |
| $\widetilde{n}_{v,c_{in}}$ | $\delta_\mu(n_{v,c_{in}})$ | $M^2(n_{v,c_{in}})$ | $M^3(n_{v,c_{in}})$ | $M^4(n_{v,c_{in}})$ | $\kappa_{n_{v,c_{in}}}$ | | |
| 26 | 5.1101 | 40.854 | $1.0338\times10^2$ | $4.9173\times10^3$ | 2.9461 | | |
| $\min n_{\aleph_v,c_{in}}$ | $\max n_{\aleph_v,c_{in}}$ | $\bar{n}_{\aleph_v,c_{in}}$ | $\sigma^2_{n_{\aleph_v,c_{in}}}$ | $\sigma_{n_{\aleph_v,c_{in}}}$ | $\bar{n}_{g,\aleph_v,c_{in}}$ | $\bar{n}_{h,\aleph_v,c_{in}}$ | $n_{\aleph_v,c_{in}}$ |
| 8 | 26 | 16.123 | 10.244 | 3.2006 | 15.809 | 15.495 | |
| $\widetilde{n}_{\aleph_v,c_{in}}$ | $\delta_\mu(n_{\aleph_v,c_{in}})$ | $M^2(n_{\aleph_v,c_{in}})$ | $M^3(n_{\aleph_v,c_{in}})$ | $M^4(n_{\aleph_v,c_{in}})$ | $\kappa_{n_{\aleph_v,c_{in}}}$ | | |
| 16 | 2.5550 | 10.214 | 12.922 | $3.0733\times10^2$ | 2.9461 | | |
| $\min n_{\aleph_e,c}$ | $\max n_{\aleph_e,c}$ | $\bar{n}_{\aleph_e,c}$ | $\sigma^2_{n,\aleph_e,c}$ | $\sigma_{n,\aleph_e,c}$ | $\bar{n}_{g,\aleph_e,c}$ | $\bar{n}_{h,\aleph_e,c}$ | $n_{\aleph_e,c}$ |
| 8 | 27 | 15.669 | 12.439 | 3.5269 | 15.279 | 14.894 | |
| $\widetilde{n}_{\aleph_e,c}$ | $\delta_\mu(n_{\aleph_e,c})$ | $M^2(n_{\aleph_e,c})$ | $M^3(n_{\aleph_e,c})$ | $M^4(n_{\aleph_e,c})$ | $\kappa_{n,\aleph_e,c}$ | | |
| 15 | 2.7792 | 12.422 | 22.761 | $4.8871\times10^2$ | 3.1673 | | |
| $\min n_{\aleph_e,c_{in}}$ | $\max n_{\aleph_e,c_{in}}$ | $\bar{n}_{\aleph_e,c_{in}}$ | $\sigma^2_{n(\aleph_e),c_{in}}$ | $\sigma_{n(\aleph_e),c_{in}}$ | $\bar{n}_{g,\aleph_e,c_{in}}$ | $\bar{n}_{h,\aleph_e,c_{in}}$ | $n_{\aleph_e,c_{in}}$ |
| 8 | 26 | 16.123 | 10.244 | 3.2006 | 15.809 | 15.495 | |
| $\widetilde{n}_{\aleph_e,c_{in}}$ | $\delta_\mu(n_{\aleph_e,c_{in}})$ | $M^2(n_{\aleph_e,c_{in}})$ | $M^3(n_{\aleph_e,c_{in}})$ | $M^4(n_{\aleph_e,c_{in}})$ | $\kappa_{n(\aleph_e),c_{in}}$ | | |
| 16 | 2.5550 | 10.214 | 12.922 | $3.0733\times10^2$ | 2.9461 | | |
| $\min n_{\aleph_f,c}$ | $\max n_{\aleph_f,c}$ | $\bar{n}_{\aleph_f,c}$ | $\sigma^2_{n(\aleph_f),c}$ | $\sigma_{n(\aleph_f),c}$ | $\bar{n}_{g,\aleph_f,c}$ | $\bar{n}_{h,\aleph_f,c}$ | $n_{\aleph_f,c}$ |
| 8 | 27 | 15.629 | 12.274 | 3.5034 | 15.242 | 14.857 | |
| $\widetilde{n}_{\aleph_f,c}$ | $\delta_\mu(n_{\aleph_f,c})$ | $M^2(n_{\aleph_f,c})$ | $M^3(n_{\aleph_f,c})$ | $M^4(n_{\aleph_f,c})$ | $\kappa_{n(\aleph_f),c}$ | | |
| 15 | 2.7635 | 12.256 | 21.051 | $4.6791\times10^2$ | 3.1149 | | |
| $\min n_{\aleph_f,c_{in}}$ | $\max n_{\aleph_f,c_{in}}$ | $\bar{n}_{\aleph_f,c_{in}}$ | $\sigma^2_{n(\aleph_f),c_{in}}$ | $\sigma_{n(\aleph_f),c_{in}}$ | $\bar{n}_{g,\aleph_f,c_{in}}$ | $\bar{n}_{h,\aleph_f,c_{in}}$ | $n_{\aleph_f,c_{in}}$ |
| 8 | 26 | 16.123 | 10.244 | 3.2006 | 15.809 | 15.495 | |
| $\widetilde{n}_{\aleph_f,c_{in}}$ | $\delta_\mu(n_{\aleph_f,c_{in}})$ | $M^2(n_{\aleph_f,c_{in}})$ | $M^3(n_{\aleph_f,c_{in}})$ | $M^4(n_{\aleph_f,c_{in}})$ | $\kappa_{n(\aleph_f),c_{in}}$ | | |
| 16 | 2.5550 | 10.214 | 12.922 | $3.0733\times10^2$ | 2.9461 | | |
| $\min A_c$ | $\max A_c$ | $\bar{A}_c$ | $\sigma^2_{A_c}$ | $\sigma_{A_c}$ | $\bar{A}g, c$ | $\bar{A}h, c$ | $A_c$ |
| 0.59209 | 83.165 | 3.4086 | 2.0373 | $2.3046\times10^2$ | 5.0726 | 1.0502 | |
| $\widetilde{A}_c$ | $\delta_\mu(A_c)$ | $M^2(A_c)$ | $M^3(A_c)$ | $M^4(A_c)$ | $\kappa_{A_c}$ | | |
| $1.1935\times10^2$ | 4.0384 | 2.8300 | 2.0530 | 1.3800 | 2.0572 | | |
| $\min A_{c_{in}}$ | $\max A_{c_{in}}$ | $\bar{A}_{c_{in}}$ | $\sigma^2_{A_{c_{in}}}$ | $\sigma_{A_{c_{in}}}$ | $\bar{A}g, c_{in}$ | $\bar{A}h, c_{in}$ | $A_{c_{in}}$ |
| 0.59209 | 3.4086 | 2.0373 | 1.0502 | 4.0384 | 2.8300 | 2.0530 | |
| $\widetilde{A}_{c_{in}}$ | $\delta_\mu(A_{c_{in}})$ | $M^2(A_{c_{in}})$ | $M^3(A_{c_{in}})$ | $M^4(A_{c_{in}})$ | $\kappa_{A_{c_{in}}}$ | | |
| 1.3800 | 2.0572 | 2.9782 | 3.9125 | 3.1031 | 1.0800 | | |
| $\min A_{fr,c}$ | $\max A_{fr,c}$ | $\bar{A}_{fr,c}$ | $\sigma^2_{A_{fr,c}}$ | $\sigma_{A_{fr,c}}$ | $\bar{A}g, fr, c$ | $\bar{A}h, fr, c$ | $A_{fr,c}$ |
| $8.5015\times10^{-8}$ | $1.1941\times10^{-5}$ | $4.8942\times10^{-7}$ | $2.9253\times10^{-7}$ | $3.3091\times10^{-5}$ | $7.2835\times10^{-7}$ | $1.5079\times10^{-7}$ | |
| $\widetilde{A}_{fr,c}$ | $\delta_\mu(A_{fr,c})$ | $M^2(A_{fr,c})$ | $M^3(A_{fr,c})$ | $M^4(A_{fr,c})$ | $\kappa_{A_{fr,c}}$ | | |
| $1.7137\times10^{-5}$ | $5.7986\times10^{-7}$ | $4.0635\times10^{-7}$ | $2.9478\times10^{-7}$ | $1.9816\times10^{-7}$ | $2.9538\times10^{-7}$ | | |
| $\min A_{fr,c_{in}}$ | $\max A_{fr,c_{in}}$ | $\bar{A}_{fr,c_{in}}$ | $\sigma^2_{A_{fr,c_{in}}}$ | $\sigma_{A_{fr,c_{in}}}$ | $\bar{A}g, fr, c_{in}$ | $\bar{A}h, fr, c_{in}$ | $A_{fr,c_{in}}$ |
| $7.5700\times10^{-4}$ | $4.3579\times10^{-3}$ | $2.6048\times10^{-3}$ | $1.3427\times10^{-3}$ | $5.1632\times10^{-3}$ | $3.6182\times10^{-3}$ | $2.6248\times10^{-3}$ | |
| $\widetilde{A}_{fr,c_{in}}$ | $\delta_\mu(A_{fr,c_{in}})$ | $M^2(A_{fr,c_{in}})$ | $M^3(A_{fr,c_{in}})$ | $M^4(A_{fr,c_{in}})$ | $\kappa_{A_{fr,c_{in}}}$ | | |
| $1.7644\times10^{-3}$ | $2.6301\times10^{-3}$ | $3.8077\times10^{-3}$ | $5.0023\times10^{-3}$ | $3.9674\times10^{-3}$ | $1.3808\times10^{-3}$ | | |
| $\min V_c$ | $\max V_c$ | $\bar{V}_c$ | $\sigma^2_{V_c}$ | $\sigma_{V_c}$ | $\bar{V}g, c$ | $\bar{V}h, c$ | $V_c$ |
| 0.88511 | $6.2052\times10^6$ | $2.7455\times10^4$ | $8.9303\times10^{10}$ | $2.9884\times10^5$ | 22.548 | 6.3296 | |
| $\widetilde{V}_c$ | $\delta_\mu(V_c)$ | $M^2(V_c)$ | $M^3(V_c)$ | $M^4(V_c)$ | $\kappa_{V_c}$ | | |
| 7.6212 | $5.1126\times10^4$ | $8.9176\times10^{10}$ | $4.3202\times10^{17}$ | $2.3781\times10^{24}$ | $2.9904\times10^2$ | | |
| $\min V_{c_{in}}$ | $\max V_{c_{in}}$ | $\bar{V}_{c_{in}}$ | $\sigma^2_{V_{c_{in}}}$ | $\sigma_{V_{c_{in}}}$ | $\bar{V}g, c_{in}$ | $\bar{V}h, c_{in}$ | $V_{c_{in}}$ |
| 0.88511 | $1.8958\times10^3$ | 20.542 | $1.5264\times10^4$ | $1.2355\times10^2$ | 5.9195 | 4.4392 | |
| $\widetilde{V}_{c_{in}}$ | $\delta_\mu(V_{c_{in}})$ | $M^2(V_{c_{in}})$ | $M^3(V_{c_{in}})$ | $M^4(V_{c_{in}})$ | $\kappa_{V_{c_{in}}}$ | | |
| 5.4355 | 27.858 | $1.5219\times10^4$ | $2.2911\times10^7$ | $3.9178\times10^{10}$ | $1.6915\times10^2$ | | |
| $\min V_{fr,c}$ | $\max V_{fr,c}$ | $\bar{V}_{fr,c}$ | $\sigma^2_{V_{fr,c}}$ | $\sigma_{V_{fr,c}}$ | $\bar{V}g, fr, c$ | $\bar{V}h, fr, c$ | $V_{fr,c}$ |
| $4.6056\times10^{-8}$ | 0.32288 | $1.4286\times10^{-3}$ | $2.4179\times10^{-4}$ | $1.5550\times10^{-2}$ | $1.1733\times10^{-6}$ | $3.2935\times10^{-7}$ | |
| $\widetilde{V}_{fr,c}$ | $\delta_\mu(V_{fr,c})$ | $M^2(V_{fr,c})$ | $M^3(V_{fr,c})$ | $M^4(V_{fr,c})$ | $\kappa_{V_{fr,c}}$ | | |
| $3.9656\times10^{-7}$ | $2.6603\times10^{-3}$ | $2.4145\times10^{-4}$ | $6.0865\times10^{-5}$ | $1.7433\times10^{-5}$ | $2.9904\times10^2$ | | |
| $\min V_{fr,c_{in}}$ | $\max V_{fr,c_{in}}$ | $\bar{V}_{fr,c_{in}}$ | $\sigma^2_{V_{fr,c_{in}}}$ | $\sigma_{V_{fr,c_{in}}}$ | $\bar{V}g, fr, c_{in}$ | $\bar{V}h, fr, c_{in}$ | $V_{fr,c_{in}}$ |
| $1.2636\times10^{-4}$ | 0.27064 | $2.9326\times10^{-3}$ | $3.1108\times10^{-4}$ | $1.7638\times10^{-2}$ | $8.4507\times10^{-4}$ | $6.3374\times10^{-4}$ | |
| $\widetilde{V}_{fr,c_{in}}$ | $\delta_\mu(V_{fr,c_{in}})$ | $M^2(V_{fr,c_{in}})$ | $M^3(V_{fr,c_{in}})$ | $M^4(V_{fr,c_{in}})$ | $\kappa_{V_{fr,c_{in}}}$ | | |
| $7.7598\times10^{-4}$ | $3.9770\times10^{-3}$ | $3.1017\times10^{-4}$ | $6.6659\times10^{-5}$ | $1.6273\times10^{-5}$ | $1.6915\times10^2$ | | |

**AVS**

The first picture I created on AVS was a Trigonal Dipyramidal. The program was the following .inp file.

```
5 2 0 0 0
1 0 0 1.2910
2 2 0 1.2910
3 1 1.7321 1.2910
4 1 0.5774 0
5 1 0.5774 2.5820
1 1 tet 1 2 3 4
2 1 tet 1 2 3 5
1 2
```

The connection of modules was ReadUCD to ExternalEdges to UViewer3D.

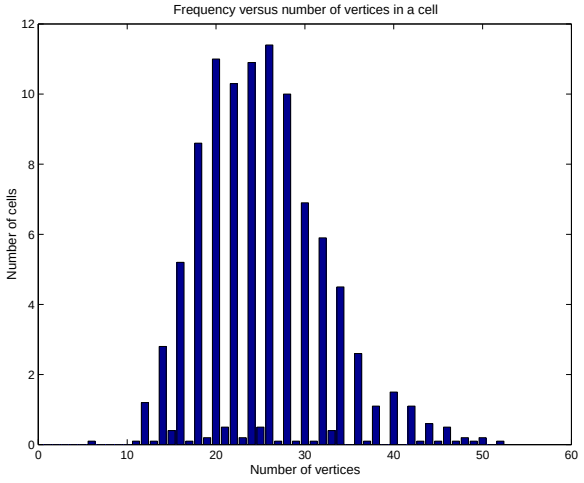
**Problems related to ISD and NQS**

There have been some problems with our Information Systems Department during the project, when Nicholas Blackaby killed about ten of my batch jobs running on Cosmos with neither warning nor justification. People in any computer department could do well if they learn how to configure the system, in this case the Network Queueing System, properly. Moreover, more communication should be encouraged among system programmers and users, so that an event like this will not cause panick and rash decision. As a consequence, I have stopped trying to simulate large networks.

**More on number of vertices 3-d**

The following was done on a 3–d systems with a joggled-input option in *qhull*. The number of vertices, however, takes into account the boundary cells as well. The minimum $n_{v,c}$ was 6, maximum $n_{v,c}$ 52, $\bar{n}_{v,c}$ 25.269, $2^{nd}$–Moment 49.363, $3^{rd}$–Moment 228.87, $4^{th}$–Moment 8,704.9, $\sigma^2_{n_{v,c}}$ 49.412, $\sigma_{n_{v,c}}$ 7.0294, $\mu_{g,n_{v,c}}$ 24.317, $\mu_{h,n_{v,c}}$ 23.365, $\widetilde{n}_{v,c}$ 24, $\delta_\mu(n_{v,c})$ 5.5428, $\kappa(n_{v,c})$ 3.5725 .

Following is a distribution graph.



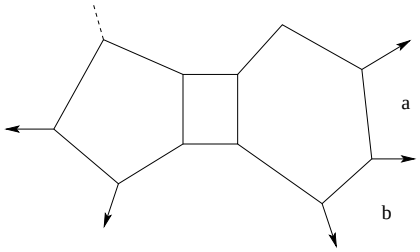

Theorem's 3.4 and 3.4 above are observations which are not only useful but essential when you try to understand the valence relations. Assuming that the Euler's theorem is true, then these two theorems give rise to Algorithm 3.2 which I have devised for counting all the components in these relations. Here $n_e$ and $n_v$ are all the edges, and respectively vertices, of the original structure originally counted, $n_b^e$ all the additional edges, going to infinity, which are drawn to complete the valences of some of the boundary vertices, $c_n$ cells with $n$ edges and $f_n$ the frequency occurences of $c_n$. The vertex at infinity mentioned does not necessarily have to be literally at the infinity. Although Algorithm 3.2 is for planes in two dimensions, it should be possible to extend it to other topolgical objects and to higher dimensions by simply changing the Euler's equation to the appropriate one. When labelling $f$, if $(n_f + n_b^f) \neq f$, we know that $(n_f + n_b^f) > f$ with probability one.

**Algorithm 3.2** *Valence relations for planes in two dimensions*

**draw** an addition bond for all boundary vertices which have valence 2**;**
$e \leftarrow \lfloor \mathbf{n_e} + \mathbf{n_b^e}/3 \rfloor$**;**
**draw** an addtional $(n_e + n_b^e - e)$ on boundary vertices**;**
**if** $\lfloor n_v/2 \rfloor \bmod 2 \neq 0$ **then**

**draw** an additional vertex at infinity;
$v \leftarrow n_v + 1$;
**else**
$v \leftarrow n_v$;
**endif**
$f \leftarrow 1 + e - v$;
**label** all the $f_n$ bound faces;
**if** $(n_f + n_b^f) = f$ **then**
**label** all the $f_b^n$ unbound faces;
**else**
**label** $(f - n_f)$ unbound faces;
**label** all the remaining $(n_f + n_b^f - f)$ unbound faces together as a single face;
**endif**

□




Paradoxically this relation holds only for infinite networks, but its derivation, as well as its verification, can only be done on a tessellation of finite size. Counting the components of the valence relation one sees in a graph involves both the Euler's theorem and Therom's 3.4 and 3.4.

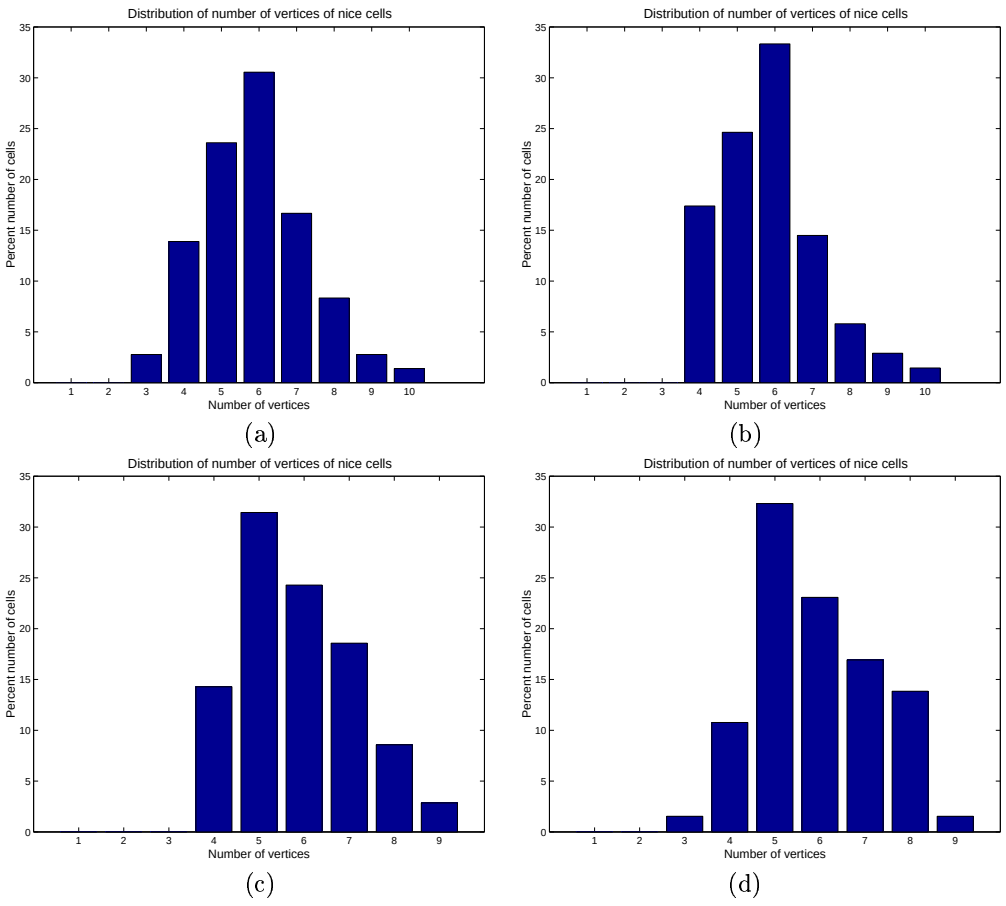
**Figure 3.35** Counting valences.

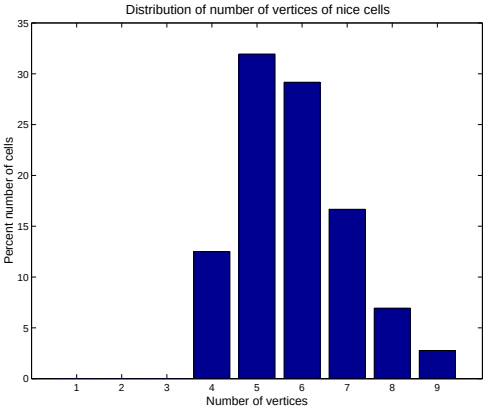
## § 3.8 Faces in different dimensions

Considering only those cells bound withing the unit box, vertices and all, the following, namely Tables 3.11, 3.12 and 3.13, are the results from five simulations in two, three and four dimensions respectively, using the program listed in § A.14.

| | **Random seed** | | | | |
|---|---|---|---|---|---|
| | 829247 | 134315 | 67453 | 432243 | 231215 |
| $N_v$ | 187 | 187 | 186 | 189 | 183 |
| $n_v$ | 170 | 167 | 168 | 163 | 170 |
| $n_c$ | 72 | 69 | 70 | 65 | 72 |
| $n_c^v$ | 169 | 165 | 166 | 159 | 170 |
| $\mu_c^v$ | 5.875 | 5.8116 | 5.8429 | 5.9077 | 5.8194 |
| $(\sigma_v^2)_c$ | 2.0264 | 1.8022 | 1.6706 | 1.7726 | 1.5022 |
| $m^2(n_c^v)$ | 1.9983 | 1.7761 | 1.6467 | 1.7453 | 1.4813 |
| $m^3(n_c^v)$ | 1.1263 | 1.6917 | 0.96591 | 0.57642 | 0.96103 |
| $n_{1f}$ | 241 | 235 | 237 | 227 | 241 |
| $n_c^{1f}$ | 240 | 233 | 235 | 223 | 241 |
| $t_{CPU}$(second) | 20.36 | 20.39 | 20.09 | 20.6 | 19.76 |

**Table 3.11** *Faces of Voronoi in two dimensions.* $N_c = 100$.


Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
(a)
Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
(b)
Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
(c)
Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
(d)

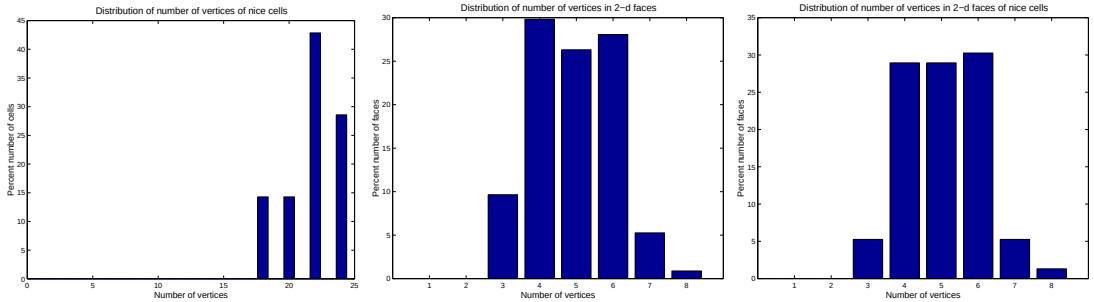



(e)

**Figure 3.37** Distribution of $v_c$ in each simulation on 2-d Voronoi..

| | **Random seed** | | | | |
|---|---|---|---|---|---|
| | 42398198 | 83250 | 34959 | 743690 | 1321 |
| $N_v$ | 204 | 221 | 224 | 225 | 214 |
| $n_v$ | 148 | 147 | 143 | 155 | 146 |
| $n_c$ | 7 | 5 | 7 | 8 | 7 |
| $n_c^v$ | 99 | 79 | 85 | 109 | 97 |
| $\mu_c^v$ | 21.714 | 23.6 | 20.286 | 21 | 21.714 |
| $(\sigma_v^2)_c$ | 4.5714 | 14.8 | 31.238 | 34.286 | 21.905 |
| $m^2(n_c^v)$ | 3.9184 | 11.84 | 26.776 | 30 | 18.776 |
| $m^3(n_c^v)$ | -4.6181 | -16.128 | -42.402 | 171 | 69.831 |
| $n_{2f}$ | 114 | 110 | 111 | 123 | 113 |
| $\mu_{2f}^v$ | 4.9211 | 4.7727 | 4.7928 | 4.8293 | 4.7788 |
| $(\sigma_v^2)_{2f}$ | 1.2592 | 1.6268 | 1.5476 | 2.0772 | 1.656 |
| $m^2(n_{2f}^v)$ | 1.2482 | 1.612 | 1.5336 | 2.0603 | 1.6413 |
| $m^3(n_{2f}^v)$ | 0.16453 | 0.99264 | 1.1334 | 1.7676 | 1.1444 |
| $n_c^{2f}$ | 76 | 60 | 69 | 88 | 75 |
| $(\mu_{2f}^v)_c$ | 5.0526 | 5.05 | 4.9855 | 4.9545 | 5.0533 |
| $(\sigma_v^2)_c^{2f}$ | 1.1439 | 1.811 | 1.3674 | 2.0209 | 1.7268 |
| $m^2((n_{2f}^v)_c)$ | 1.1288 | 1.7808 | 1.3476 | 1.9979 | 1.7038 |
| $m^3((n_{2f}^v)_c)$ | 0.19004 | 0.28275 | 1.5224 | 1.4544 | 0.98057 |
| $n_c^{1f}$ | 167 | 133 | 146 | 187 | 163 |
| $t_{CPU}$(second) | 24.28 | 25.67 | 28.49 | 34.23 | 26.46 |

**Table 3.12** *Various faces of Voronoi in three dimensions.* $N_c = 50$.


Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
Distribution of number of vertices in 2−d faces
Percent number of faces
Number of vertices
Distribution of number of vertices in 2−d faces of nice cells
Percent number of faces
Number of vertices

(a1) (b1) (c1)

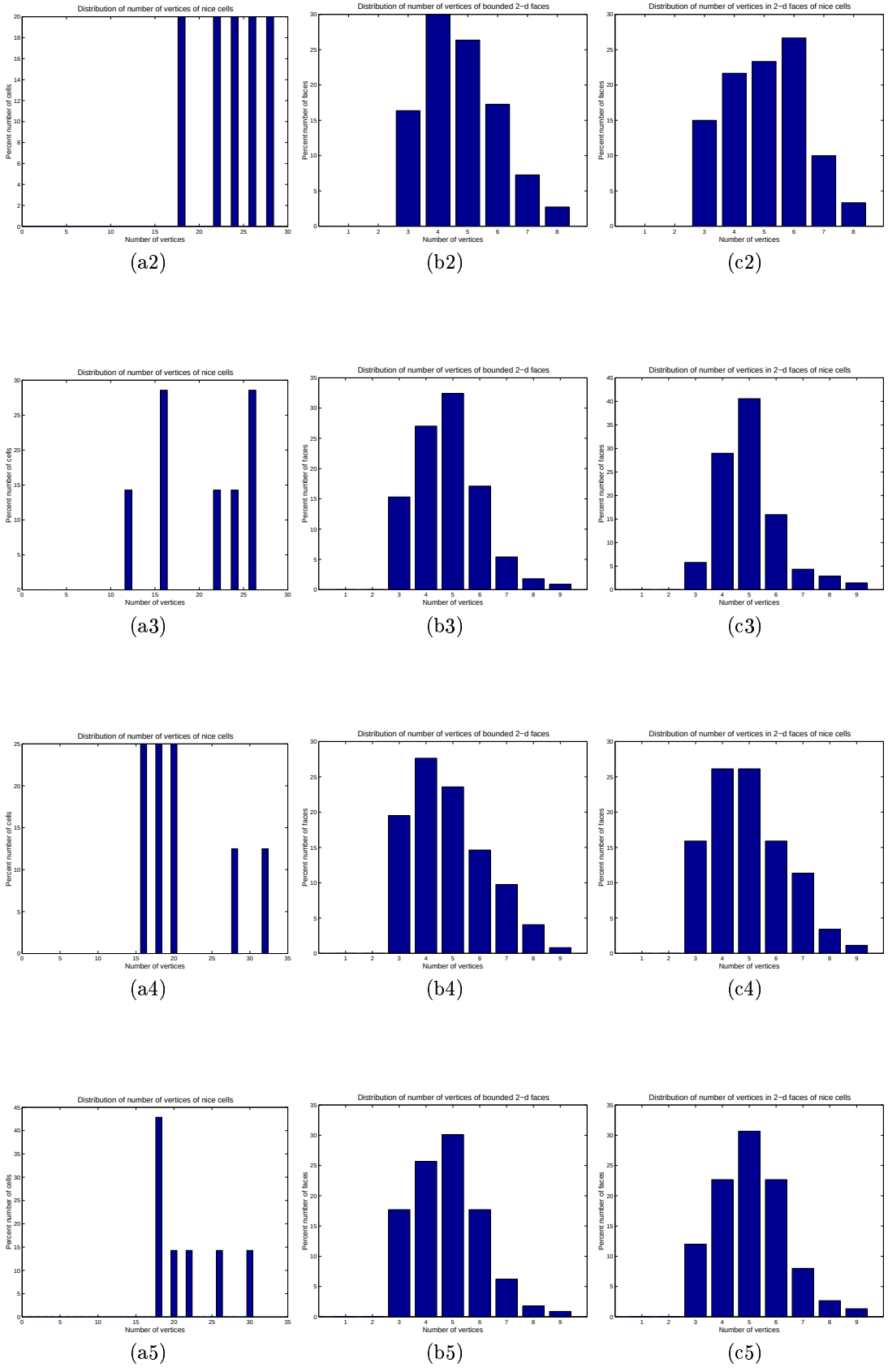
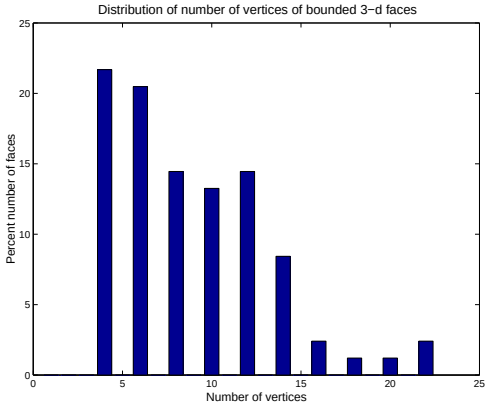



**Figure 3.38** *Distribution of* (ai) $v_c$, (bi) $v_{2\mathrm{f}}$, *and* (ci) $v_c^{2\mathrm{f}}$ *of the* $i^{\mathrm{th}}$ *simulation on 3-d Voronoi.*

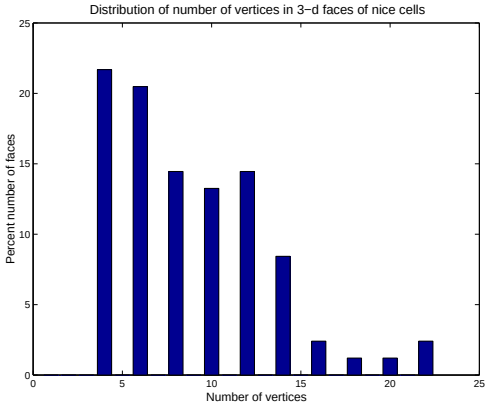
| | **Random seed** | | | | |
|---|---|---|---|---|---|
| | 24098 | 802723 | 1453 | 732849 | 20480 |
| $N_v$ | 1684 | 1719 | 1674 | 1586 | 1600 |
| $n_v$ | 938 | 921 | 860 | 889 | 904 |
| $n_c$ | 5 | 1 | 1 | 1 | 2 |
| $n_c^v$ | 442 | 172 | 149 | 117 | 200 |
| $\mu_c^v$ | 114 | 172 | 149 | 117 | 110 |
| $(\sigma_v^2)_c$ | 174.5 | 0 | 0 | 0 | 72 |
| $m^2(n_c^v)$ | 139.6 | 0 | 0 | 0 | 36 |
| $m^3(n_c^v)$ | -547.2 | 0 | 0 | 0 | 0 |
| $n_{3f}$ | 83 | 102 | 84 | 83 | 95 |
| $\mu_{3f}^v$ | 8.8675 | 8.3137 | 8.5952 | 8.9398 | 8.7789 |
| $(\sigma_v^2)_{3f}$ | 18.848 | 18.198 | 19.449 | 17.496 | 19.77 |
| $m^2(n_{3f}^v)$ | 18.621 | 18.019 | 19.217 | 17.286 | 19.562 |
| $m^3(n_{3f}^v)$ | 75.225 | 102.38 | 95.854 | 62.919 | 118.61 |
| $n_c^{3f}$ | 32 | 8 | 10 | 7 | 19 |
| $(\mu_{3f}^v)_c$ | 8.8675 | 9.25 | 10.8 | 11.143 | 9.8947 |
| $(\sigma_v^2)_c^{3f}$ | 18.848 | 13.643 | 21.511 | 14.476 | 28.655 |
| $m^2((n_{3f}^v)_c)$ | 18.621 | 11.938 | 19.36 | 12.408 | 27.147 |
| $m^3((n_{3f}^v)_c)$ | 75.225 | 18.281 | 66.624 | 8.5364 | 195.1 |
| $n_c^{2f}$ | 11 | 0 | 5 | 2 | 7 |
| $n_c^{1f}$ | 2 | 0 | 0 | 0 | 1 |
| $t_{CPU}$(second) | 315.07 | 358.94 | 324.56 | 281.85 | 283.82 |

**Table 3.13** *Faces of Voronoi in four dimensions.* $N_c = 100$.

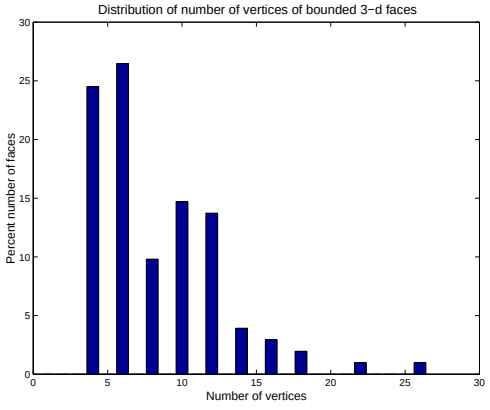
The number of vertices in each 2-d face is a constant equals to three while that of a 1-d face is two. In the first run the number of vertices in each of the cells is 95, 108, 117, 120 and 130; in the last run, this number is 104 and 116.



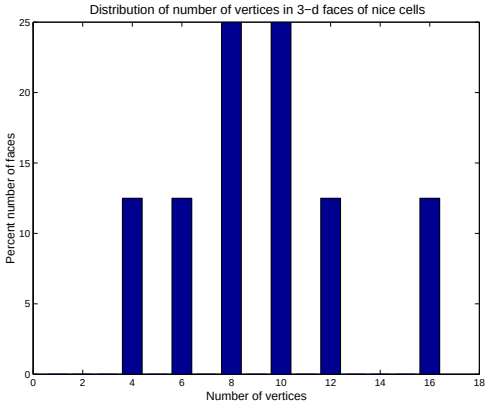

(a1)




(b1)




(a2)




(b2)

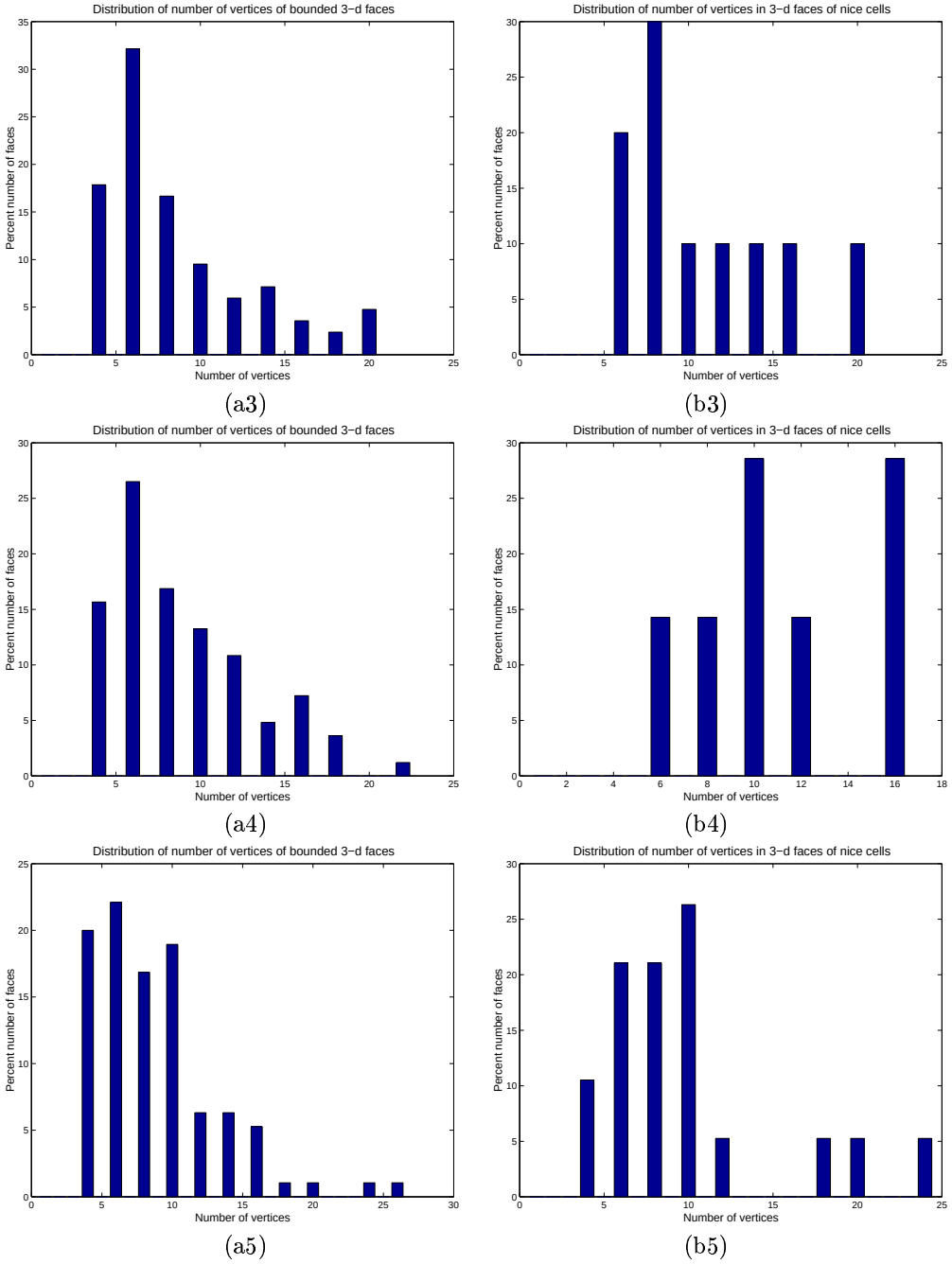
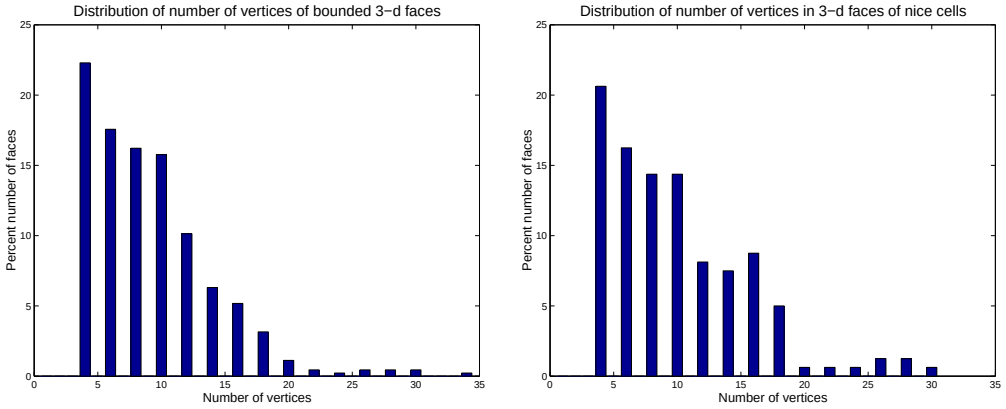

Distribution of number of vertices of bounded 3−d faces
Percent number of faces
Number of vertices
(a3)
Distribution of number of vertices in 3−d faces of nice cells
Percent number of faces
Number of vertices
(b3)
Distribution of number of vertices of bounded 3−d faces
Percent number of faces
Number of vertices
(a4)
Distribution of number of vertices in 3−d faces of nice cells
Percent number of faces
Number of vertices
(b4)
Distribution of number of vertices of bounded 3−d faces
Percent number of faces
Number of vertices
(a5)
Distribution of number of vertices in 3−d faces of nice cells
Percent number of faces
Number of vertices
(b5)

**Figure 3.39** *Distribution of* (ai) $v_{3\mathrm{f}}$ *and* (bi) $v_c^{3\mathrm{f}}$ *of the* $i^{\mathrm{th}}$ *simulation on 4-d Voronoi.*

Table 3.14 contains the results obtained from a 4-d Voronoi network of 300 original nuclei. The numbers of vertices of the sixteen cells are 97, 99, 112, 141, 145, 160, 170, 171, 176, 176, 184, 186, 188, 192, 216 and 235.

| **Random seed** | 91876 | | | | |
|---|---|---|---|---|---|
| $\mathrm{N}_v$ | 6776 | $m^3(n_c^v)$ | $-1.7507 \times 10^4$ | $(\mu_{3\mathrm{f}}^v)_c$ | 9.9875 |
| $n_v$ | 3848 | $n_{3\mathrm{f}}$ | 444 | $(\sigma_v^2)_c^{3\mathrm{f}}$ | 31.1697 |
| $n_c$ | 16 | $\mu_{3\mathrm{f}}^v$ | 9.1486 | $m^2((n_{3\mathrm{f}}^v)_c)$ | 30.9748 |
| $n_c^v$ | 1687 | $(\sigma_v^2)_{3\mathrm{f}}$ | 24.0411 | $m^3((n_{3\mathrm{f}}^v)_c)$ | 203.3116 |
| $\mu_c^v$ | 165.5000 | $m^2(n_{3\mathrm{f}}^v)$ | 23.9869 | $n_c^{2\mathrm{f}}$ | 60 |
| $(\sigma_v^2)_c$ | $1.5100 \times 10^3$ | $m^3(n_{3\mathrm{f}}^v)$ | 170.4755 | $n_c^{1\mathrm{f}}$ | 3 |
| $m^2(n_c^v)$ | $1.4156 \times 10^3$ | $n_c^{3\mathrm{f}}$ | 160 | $\mathrm{t_{CPU}(second)}$ | $1.6013 \times 10^4$ |

**Table 3.14** *Face statistics of 300 nuclei Voronoi in four dimensions.*


Distribution of number of vertices of bounded 3−d faces
Percent number of faces
Number of vertices
Distribution of number of vertices in 3−d faces of nice cells
Percent number of faces
Number of vertices

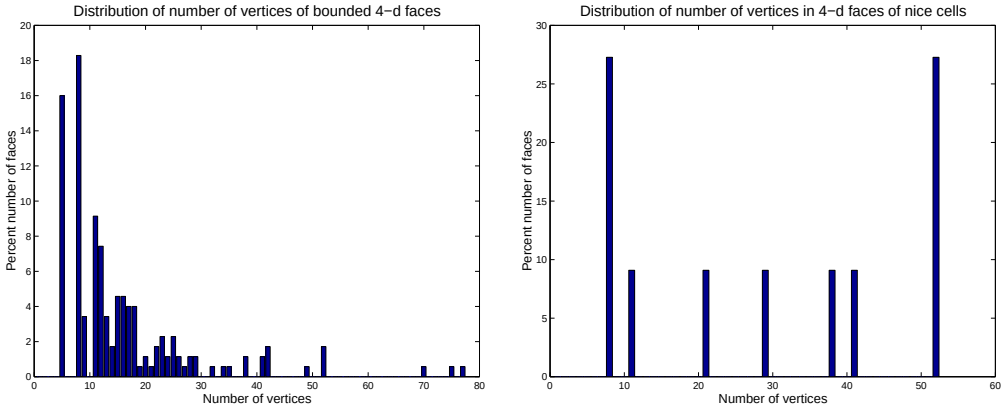
(a) (b)

**Figure 3.40** *Distribution of* (a) $v_{3\mathrm{f}}$ *and* (bi) $v_c^{3\mathrm{f}}$ *of 4-d Voronoi, 300 nuclei.*

Table 3.15 is obtained from Voronoi in five dimensions.

| **Random seed** | 39378 | | | $\mathrm{N}_v$ | 16212 |
|---|---|---|---|---|---|
| $n_v$ | 6449 | $n_{4\mathrm{f}}$ | 175 | $(\sigma_v^2)_c^{4\mathrm{f}}$ | 352.2909 |
| $n_c$ | 1 | $\mu_{4\mathrm{f}}^v$ | 15.9200 | $m^2((n_{4\mathrm{f}}^v)_c)$ | 320.2645 |
| $n_c^v$ | 864 | $(\sigma_v^2)_{4\mathrm{f}}$ | 163.7752 | $m^3((n_{4\mathrm{f}}^v)_c)$ | 351.8362 |
| $\mu_c^v$ | 864 | $m^2(n_{4\mathrm{f}}^v)$ | 162.8393 | $n_c^{3\mathrm{f}}$ | 1 |
| $(\sigma_v^2)_c$ | 0 | $m^3(n_{4\mathrm{f}}^v)$ | $4.8764 \times 10^3$ | $(\mu_{3\mathrm{f}}^v)_c$ | 4 |
| $m^2(n_c^v)$ | 0 | $n_c^{4\mathrm{f}}$ | 11 | $n_c^{2\mathrm{f}}$ | 1 |
| $m^3(n_c^v)$ | 0 | $(\mu_{4\mathrm{f}}^v)_c$ | 29.0909 | $\mathrm{t_{CPU}}$(second) | $1.8908 \times 10^4$ |

**Table 3.15** *Face statistics of 200 nuclei Voronoi in five dimensions.*

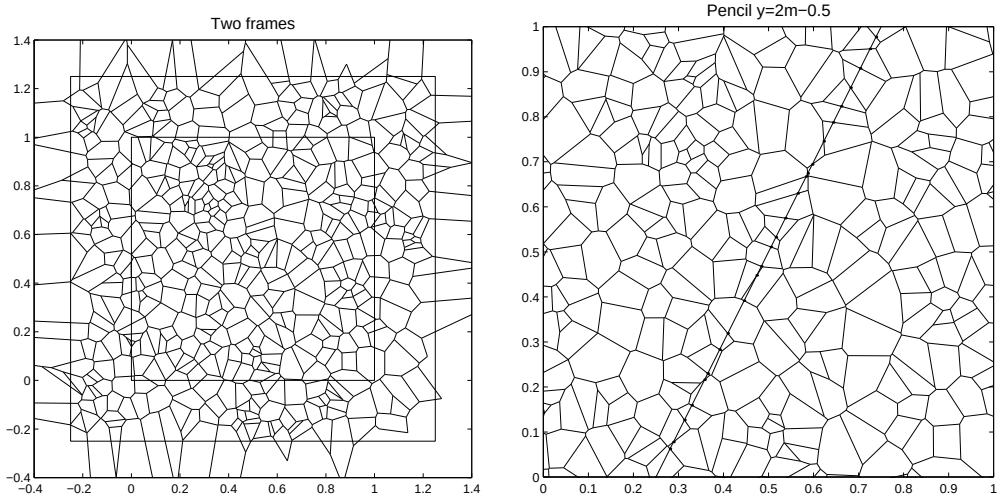

Distribution of number of vertices of bounded 4−d faces
Percent number of faces
Number of vertices
Distribution of number of vertices in 4−d faces of nice cells
Percent number of faces
Number of vertices

(a) (b)

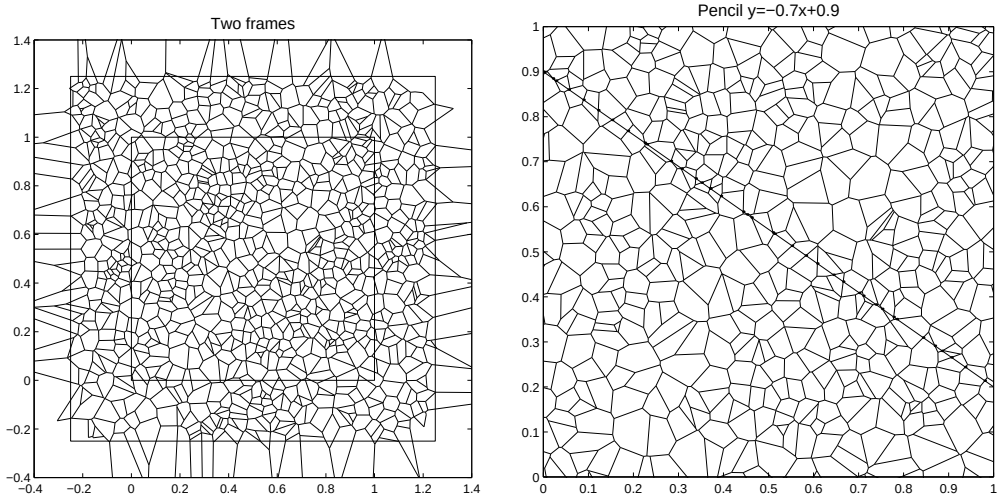
**Figure 3.41** *Distribution of* (a) $v_{4\mathrm{f}}$ *and* (bi) $v_c^{4\mathrm{f}}$ *of 5-d Voronoi, 200 nuclei.*

## § 3.9 Beam intersection study

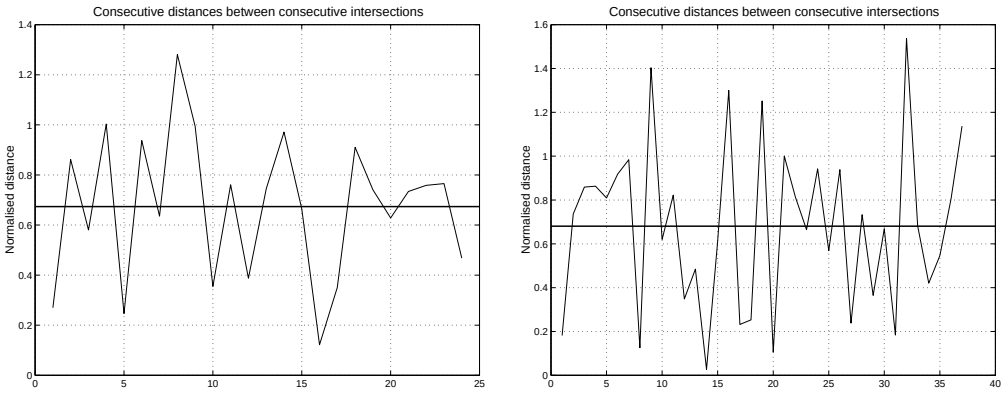
In this study of sectioning by a line the Voronoi in two dimensions, first generates on Matlab 500 points within a square box from $-0.25$ to $1.25$ in both axes. The beam is simply a straight line $y = mx + c$ where $m$ is the slope and $c$ a constant.




(a) (b)




(c) (d)




(e) (f)

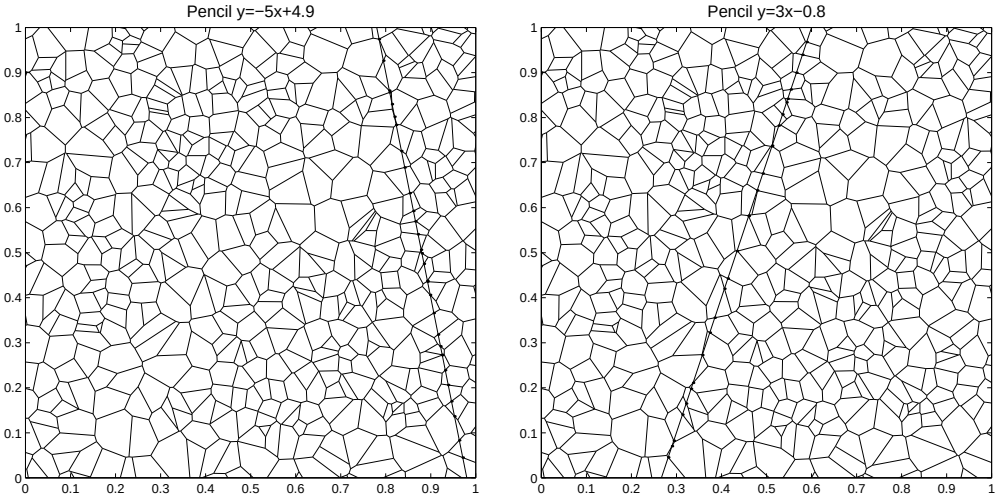



**Figure 3.16** *Intersection by a line.* (a) *is intersected by* (b) $y = 2x - 0.5$*;* (c) *is intersected by* (d) $y = -0.7x + 0.9$*;* (e) *and* (f) *are corresponding distances of respectively* (b) *and* (d)*;* (c) *is intersected by* (g) $y = -5x + 4.9$ *and* (h) $y = 3x - 0.8$.

A natural basis for the normalisation is the expected distance. Another possible basis is $\frac{1}{\sqrt{239}} = 0.064685$. I call *mean normalisation* the normalisation using the first basis and *homogeneity normalisation* the second. Graphs of the closest pair distances look the same for both types of normalisation.

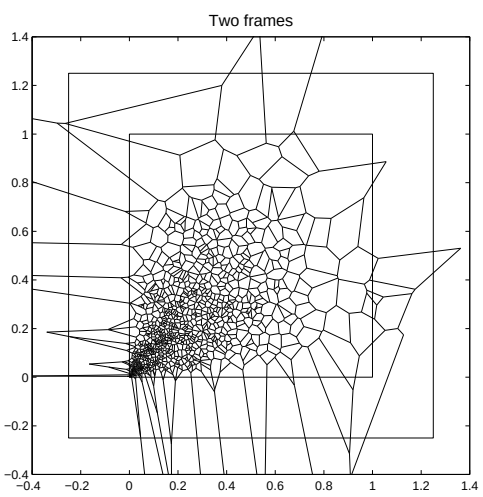
| **Simulation** | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $N_c$ | 500 | 1,000 | | |
| $N_e$ | 1,463 | 2,955 | | |
| $n_c$ | 239 | 463 | | |
| $n_e$ | 789 | 1,455 | | |
| Line equation | $y = 2x - 0.5$ | $y = -0.7x + 0.9$ | $y = -5x + 4.9$ | $y = 3x - 0.8$ |
| $\bar{d}$ | $4.6867 \times 10^{-2}$ | $3.163 \times 10^{-2}$ | $3.2448 \times 10^{-2}$ | $3.7843 \times 10^{-2}$ |
| $\sigma_d^2$ | $3.9422 \times 10^{-4}$ | $3.1174 \times 10^{-4}$ | $2.9005 \times 10^{-4}$ | $4.9535 \times 10^{-4}$ |
| Normalisation | | | | |
| by mean | $1 \pm \substack{0.4237 \\ 0.1713 \\ -8.1332\times10^{-3}}$ | $1 \pm \substack{0.5582 \\ 0.3032 \\ 3.5783\times10^{-2}}$ | $1 \pm \substack{0.5249 \\ 0.2656 \\ -3.7762\times10^{-3}}$ | $1 \pm \substack{0.5881 \\ 0.3321 \\ 2.2989\times10^{-2}}$ |
| by homogeneity | $0.7245 \pm \substack{0.3070 \\ 8.9936\times10^{-2} \\ -3.0936\times10^{-3}}$ | $0.68059 \pm \substack{0.3799 \\ 0.1404 \\ 1.128\times10^{-2}}$ | $0.6982 \pm \substack{0.3665 \\ 0.1295 \\ -1.2853\times10^{-3}}$ | $0.81428 \pm \substack{0.4789 \\ 0.2202 \\ 1.2412\times10^{-2}}$ |

The program used is listed in § A.15. The space position vector of the intersection between the two vectors $AB$ and $CD$ is $P = A + r(B - A)$, where $AB = B - A$ and $CD = D - C$ are vectors or directed lines and $A, B, C, D$ are space vectors, $r = \begin{vmatrix} CD' \\ CA' \end{vmatrix} \Big/ \begin{vmatrix} AB' \\ CD' \end{vmatrix}$ and $s = \begin{vmatrix} AB' \\ CA' \end{vmatrix} \Big/ \begin{vmatrix} AB' \\ CD' \end{vmatrix}$ Here $AB = A + r(B - A)$, and $CD = C + s(D - C), 0 \le r, s \le 1$ are directed lines. $P$ exists if $0 \le r \le 1$ and $0 \le s \le 1$.

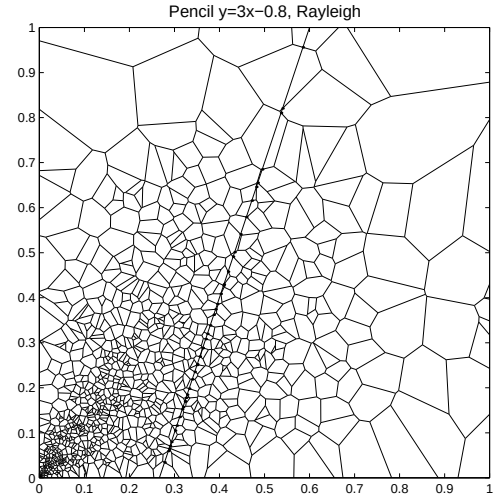
If the denominator $\begin{vmatrix} AB' \\ CD' \end{vmatrix}$ is zero, then the two lines are parallel. Also, if the nominator of $r$ is zero, that is $\begin{vmatrix} CD' \\ CA' \end{vmatrix} = 0$, then both lines are collinear.

Consider the line section of Rayleigh distributed Voronoi where both the coordinates $x$ and $y$ are random numbers with Rayleigh distribution. The probability density function of the Rayleigh distribution is $y = f(x/b) = \frac{x}{b^2} e^{\frac{-x^2}{2b^2}}$. The mean of this distribution is $b\sqrt{\frac{\pi}{2}}$ and the variance is $\frac{4-\pi}{2} b^2$. With $b = 1, 2, \ldots, 1000$ choose the random numbers from the Rayleigh distribution, then scale and use them as the coordinates.

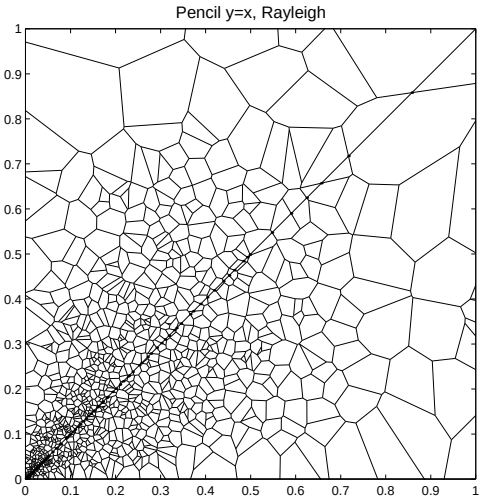
In Figure 3.44 the structure in (a) is intersected by (b) $y = 3x - 0.8$, (c) $y = x$, (d) $y = 2x$, (e) $y = 0.2x$, (f) $y = -x + 1$, (g) $y = -x + 0.3$.


Two frames

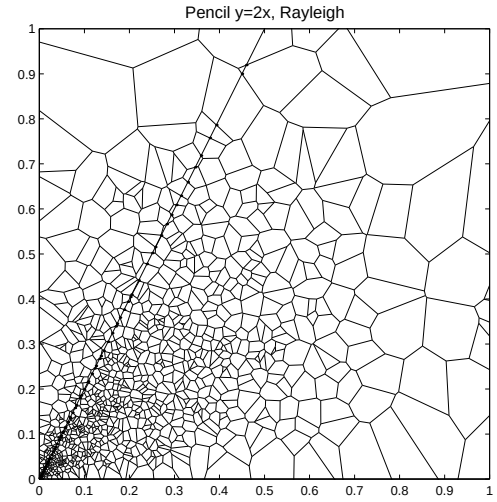
(a)


Pencil y=3x−0.8, Rayleigh

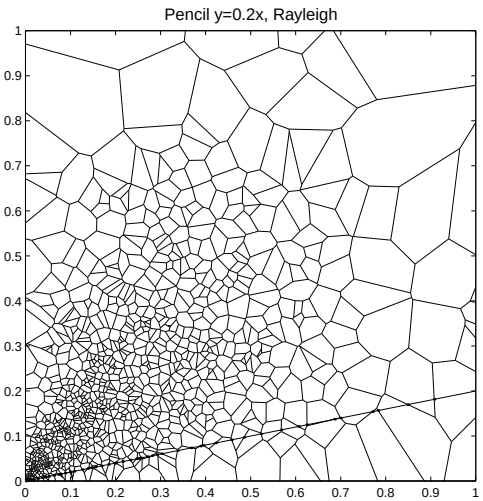
(b)


Pencil y=x, Rayleigh

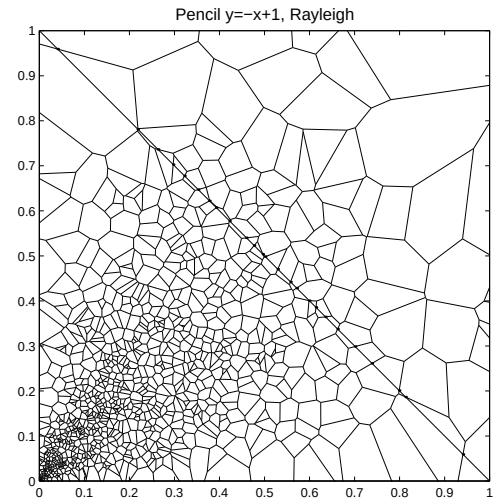
(c)


Pencil y=2x, Rayleigh

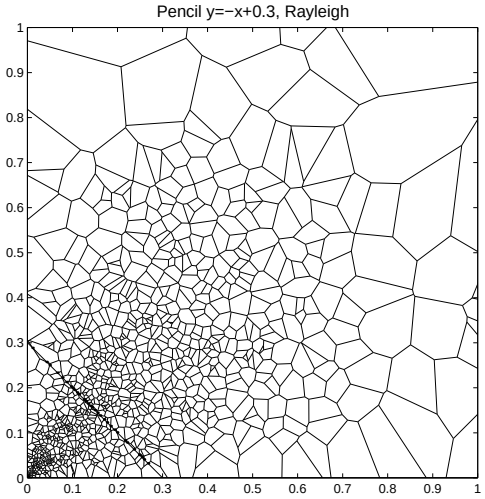
(d)


Pencil y=0.2x, Rayleigh

(e)


Pencil y=−x+1, Rayleigh

(f)




(g)

**Figure 3.44** Line intersection in Rayleigh distributed Voronoi..

| **Simulation** | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| $N_c$ | 1,000 | | | | | |
| $N_e$ | 2,975 | | | | | |
| $n_c$ | 988 | | | | | |
| $n_e$ | 2,912 | | | | | |
| Line eq. | $y = 3x - 0.8$ | $y = x$ | $y = 2x$ | $y = 0.2x$ | $y = -x + 1$ | $y = -x + 0.3$ |
| $\bar{d}$ | $2.3621 \times 10^{-2}$ | $1.4253 \times 10^{-2}$ | $1.3853 \times 10^{-2}$ | $1.7434 \times 10^{-2}$ | $4.4401 \times 10^{-2}$ | $1.027 \times 10^{-2}$ |
| $\sigma_d^2$ | $8.1582 \times 10^{-4}$ | $6.2922 \times 10^{-4}$ | $2.9647 \times 10^{-4}$ | $3.0991 \times 10^{-4}$ | $1.1584 \times 10^{-3}$ | $3.5365 \times 10^{-5}$ |
| Normalised | | | | | | |
| by mean | $1 \pm 1.2092$<br>1.4256<br>5.5516 | $1 \pm 1.7599$<br>3.0609<br>28.351 | $1 \pm 1.2429$<br>1.5239<br>7.6499 | $1 \pm 1.0098$<br>1.0004<br>1.6643 | $1 \pm 0.7665$<br>0.56204<br>1.2114 | $1 \pm 0.5791$<br>0.3245<br>0.1755 |
| by homog. | $0.7425 \pm 0.8978$<br>0.7859<br>2.2721 | $0.4480 \pm 0.7885$<br>0.6144<br>2.5493 | $0.4354 \pm 0.5412$<br>0.2890<br>0.6316 | $0.548 \pm 0.5533$<br>0.3004<br>0.2739 | $1.3956 \pm 1.0698$<br>1.0947<br>3.293 | $0.3228 \pm 0.1869$<br>$3.3813 \times 10^{-2}$<br>$5.9021 \times 10^{-3}$ |

**Table 3.17** *Line intersection statistics of Rayleigh distributed Voronoi*

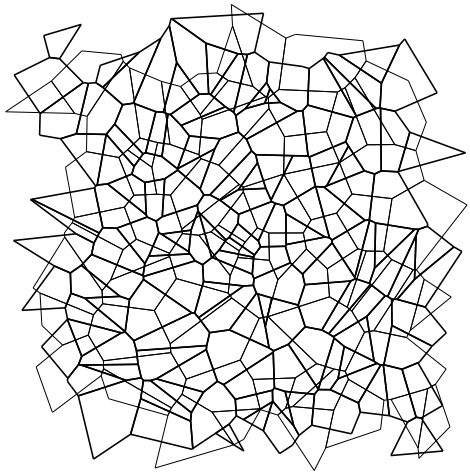
The following codes generate and scale the points of Rayleigh distribution.

```
X=raylrnd([1:NumCell])';Y=raylrnd([1:NumCell])';Max=0.8*max([X;Y]);X=X/Max;Y=Y/Max;
```

The values of $Cx$ and $Dx$ will need to be adjusted manually from the pencil beam equation. It is the value of $x$ at both points of intersection between the beam and the $[0, 1]$ square box. A random Voronoi network can either be both homogeneous and isotropic or, nonhomogeneous and nonisotropic depending on whether the probability distribution function is a constant.

## § 3.10 Voronoi of Voronoi

Voronoi of Voronoi is still a Voronoi. But what are the effects of applying the Voronoi tessellating operator on a set of points? It is an interesting question to consider whether Voronoi tessellation as an operator alters, for instance, the nature of the distribution of the original set of points. Figure 3.46 shows the first Voronoi operator applied to the original points and then the second Voronoi operator applied to the vertices obtained.



In Figure the Voronoi operator is twiced in succession, *i.e.* $\mathcal{V}^2(\cdot)$, starting from a set of 100 points.

**Figure 3.46** Applying the Voronoi operator twice in succession.

Let $\mathcal{V}^n(\cdot)$ be the $n^{\text{th}}$-order Voronoi operator. Figure 3.46 shows as an example $\mathcal{V}^2(x)$ where $x$ is the original generator points. In Figure 3.47 this operator is applied six times in succession. The Voronoi operator is a mapping which maps a set of points into a set of Voronoi vertices having the former points as the Voronoi nuclei. The program used for Figures 3.46 to 3.49 is listed in § A.18.:

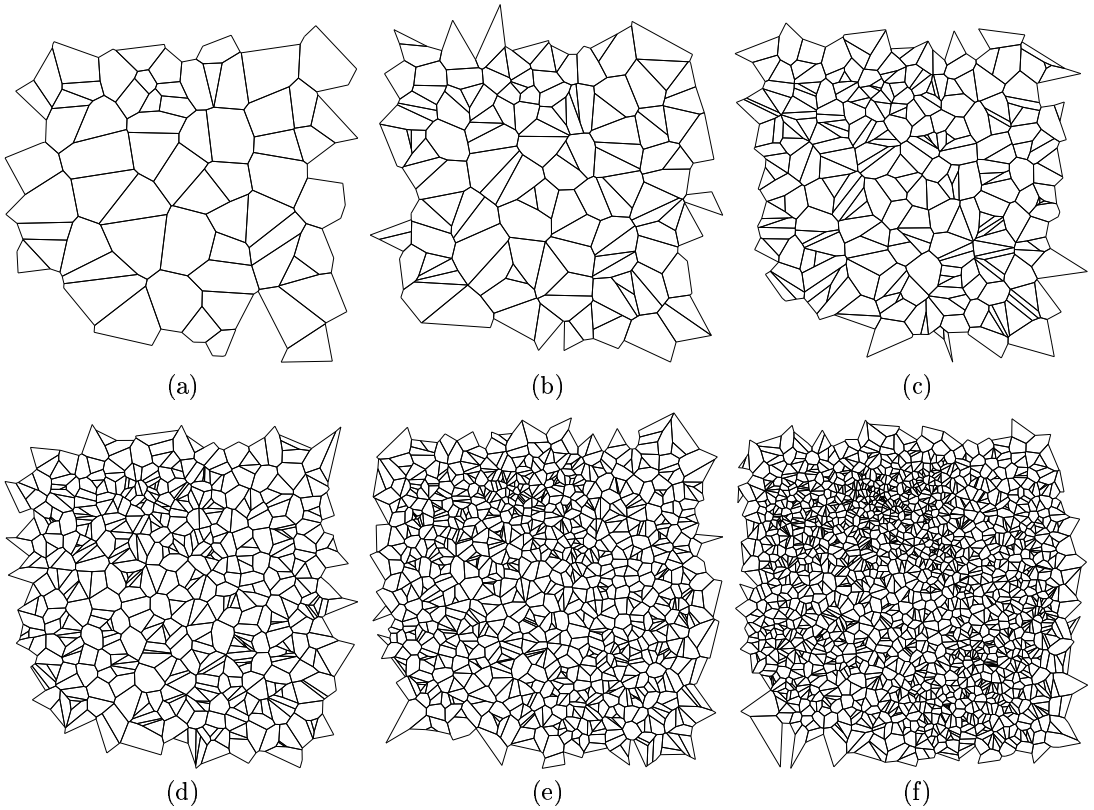



**Figure 3.47** *The Voronoi operations,* (a) $\mathcal{V}(x)$, (b) $\mathcal{V}^2(x)$, (c) $\mathcal{V}^3(x)$, (d) $\mathcal{V}^4(x)$, (e) $\mathcal{V}^5(x)$, (f) $\mathcal{V}^6(x)$, *where* $x$ *is the original set of 100 points.*

Figure 3.49 shows the effect produced by the Voronoi operators of various degrees on the number of cells, $n_c$, and vertices, $n_v$ of the network.

The number of cells and vertices when recursively applying the Voronoi operators of various orders on the set of point $x$ is shown in Figure 3.49, that is to say, $\mathcal{V}(x)$, $\mathcal{V}^2(x)$, $\cdots$, $\mathcal{V}^6(x)$. From Figure 3.49, the increase of $n_c$ and $n_v$ with the order $n$ of $\mathcal{V}^n(\cdot)$ is exponential. And from Figure 3.49 (a) to (f), apart from the unevenness affected at the boundary, the grephs shows the degree of lumpiness in the original distribution retained by the Voronoi operator. This is analogous to the increase in the entropy in physical processes.

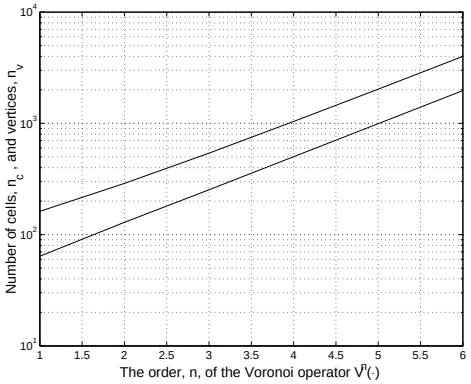
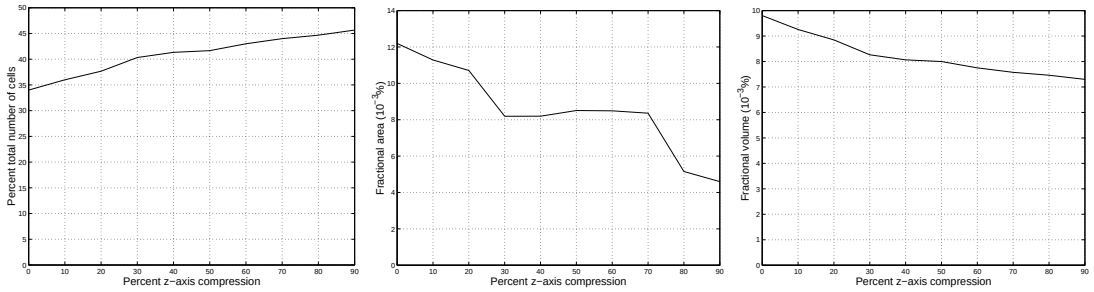



**Figure 3.49** Recursively applying the Voronoi operators of various orders.

## § 3.11 Transformations of Voronoi

The simplest of transformations is compression, where the coordinates of each vertex in multiplied by a factor less than one. Figure 3.50 shows, within a fixed box, the effect the compression along the $z$-axis has on the number of cells, surface area and volume of cells within the box.

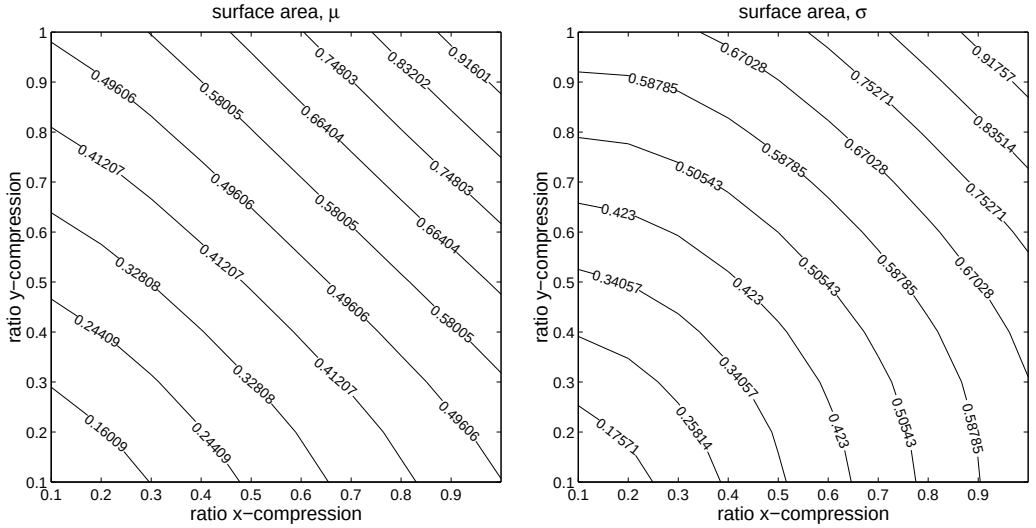



(a) (b) (c)

**Figure 3.50** *The effect of z-axis compression on* (a) *the number of cells in the box,* (b) *the surface area of a cell and* (c) *the volume of a cell.*

## § 3.12 Compressed Voronoi

The following study looks at compression of the Voronoi tessellation in three dimensions. With the compression simultaneously in the $x$- and $y$ axes, the mean and standard deviation of cell's surface area are shown as contours in Figure 3.51. The numerical values are shown in Figure's 3.18 and 3.19. Here $x_i$ and $y_i$ are respectively transformed to $r_x x_i$ and $r_y y_i$. All the values here are normalised by their corresponding values in the case without compression. In this case, the network was created from 100 generators within one unit cube, only 21 inner cells are considered, the mean and the standard deviation of the cell surface area are respectively 0.2330 and 0.0544.




(a) (b)

**Figure 3.51** (a) *Mean, and* (b) *standard deviation of surface area of cells under compression, normalised against the corresponding values of the uncompressed case. In other words,* (a) $\mu(a_{ij})/\mu(a_{11})$, *and* (b) $\sigma(a_{ij})/\sigma(a_{11})$.

$\boxed{\mu(a_{ij})}$

| | | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| | **1.0** | 1.0000 | 0.9337 | 0.8688 | 0.8058 | 0.7450 | 0.6870 | 0.6326 | 0.5828 | 0.5396 | 0.5060 |
| | **0.9** | 0.9318 | 0.8683 | 0.8061 | 0.7457 | 0.6873 | 0.6315 | 0.5790 | 0.5310 | 0.4892 | 0.4566 |
| | **0.8** | 0.8650 | 0.8042 | 0.7447 | 0.6866 | 0.6305 | 0.5768 | 0.5261 | 0.4797 | 0.4391 | 0.4073 |
| | **0.7** | 0.8001 | 0.7419 | 0.6847 | 0.6290 | 0.5749 | 0.5230 | 0.4740 | 0.4289 | 0.3893 | 0.3581 |
| | **0.6** | 0.7375 | 0.6816 | 0.6267 | 0.5730 | 0.5208 | 0.4706 | 0.4230 | 0.3789 | 0.3400 | 0.3092 |
| $r_y$ | **0.5** | 0.6777 | 0.6240 | 0.5711 | 0.5193 | 0.4687 | 0.4199 | 0.3734 | 0.3300 | 0.2914 | 0.2605 |
| | **0.4** | 0.6217 | 0.5699 | 0.5188 | 0.4685 | 0.4193 | 0.3715 | 0.3257 | 0.2826 | 0.2439 | 0.2123 |
| | **0.3** | 0.5707 | 0.5205 | 0.4708 | 0.4217 | 0.3735 | 0.3264 | 0.2808 | 0.2375 | 0.1979 | 0.1648 |
| | **0.2** | 0.5269 | 0.4779 | 0.4293 | 0.3810 | 0.3333 | 0.2863 | 0.2404 | 0.1962 | 0.1547 | 0.1188 |
| | **0.1** | 0.4939 | 0.4457 | 0.3976 | 0.3497 | 0.3020 | 0.2546 | 0.2078 | 0.1617 | 0.1171 | 0.0761 |

**Table 3.18** *Mean of the cell surface area of compressed Voronoi*

$\boxed{\sigma(a_{ij})}$

| | | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| | **1.0** | 1.0000 | 0.9381 | 0.8789 | 0.8231 | 0.7715 | 0.7255 | 0.6868 | 0.6577 | 0.6407 | 0.6382 |
| | **0.9** | 0.9360 | 0.8761 | 0.8186 | 0.7641 | 0.7135 | 0.6678 | 0.6288 | 0.5986 | 0.5799 | 0.5751 |
| | **0.8** | 0.8756 | 0.8173 | 0.7611 | 0.7075 | 0.6573 | 0.6116 | 0.5720 | 0.5403 | 0.5195 | 0.5122 |
| | **0.7** | 0.8199 | 0.7626 | 0.7071 | 0.6539 | 0.6037 | 0.5574 | 0.5165 | 0.4830 | 0.4596 | 0.4495 |
| | **0.6** | 0.7699 | 0.7130 | 0.6576 | 0.6041 | 0.5532 | 0.5058 | 0.4631 | 0.4271 | 0.4005 | 0.3870 |
| $r_y$ | **0.5** | 0.7269 | 0.6696 | 0.6136 | 0.5592 | 0.5069 | 0.4576 | 0.4123 | 0.3731 | 0.3426 | 0.3249 |
| | **0.4** | 0.6924 | 0.6340 | 0.5766 | 0.5205 | 0.4661 | 0.4141 | 0.3654 | 0.3219 | 0.2864 | 0.2634 |
| | **0.3** | 0.6676 | 0.6076 | 0.5482 | 0.4897 | 0.4324 | 0.3769 | 0.3240 | 0.2751 | 0.2331 | 0.2030 |
| | **0.2** | 0.6536 | 0.5914 | 0.5296 | 0.4682 | 0.4076 | 0.3481 | 0.2902 | 0.2350 | 0.1848 | 0.1451 |
| | **0.1** | 0.6495 | 0.5853 | 0.5212 | 0.4573 | 0.3936 | 0.3302 | 0.2675 | 0.2059 | 0.1466 | 0.0933 |

**Table 3.19** *Standard deviation of the cell surface area of compressed Voronoi*

Mistakes reveal some interesting characteristics worth investigating further, namely those shown in Figure 3.52.

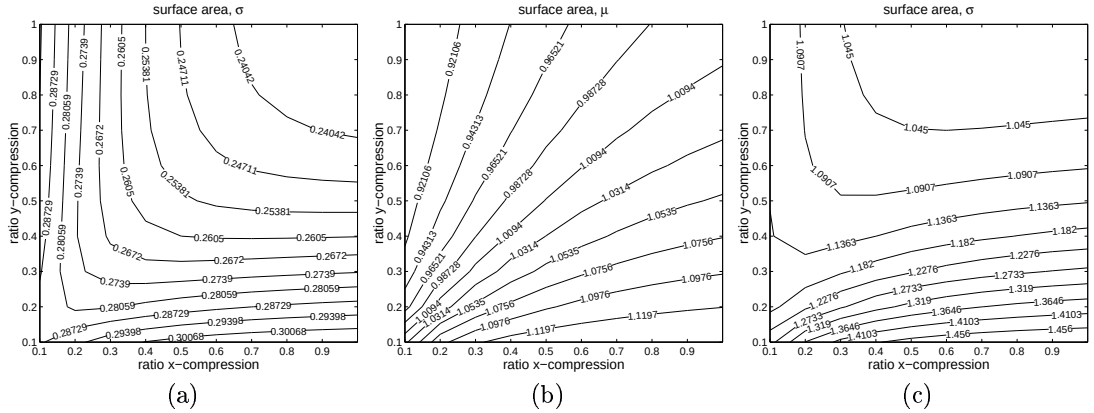



**Figure 3.52** *Some hidden characteristics revealed from out choice of the normalisation basis;* (a) $\sigma(a_{ij}/\mu(a_{ij}))$, (b) $\mu(a_{ij}/a_{ij}(1))/\mu(a_{11}/a_{11}(1))$ *and* (c) $\sigma(a_{ij}/a_{ij}1())/\sigma(a_{11}/a_{11}(1))$

Let $a_{ij}$ be the surface area of cells when subjected to compressions $i$ and $j$ respectively along the $x$- and $y$ axes, that is to say, $r_x = i$ and $r_y = j$. And let $a_{ij}(k)$ be that surface area of the $k^{\text{th}}$ cell among those undergoing these compressions $ij$. Then Figure 3.52 (b) and (c) are possible when we pick one cell as the basis for our normalisation, while Figure 3.52 (a) is when we normalise each compressed case by its own mean.

For cell volume of the same network the results are shown in Figure 3.53. The mean value and the standard deviation of cell volumes are respectively 0.0075 and 0.0032.

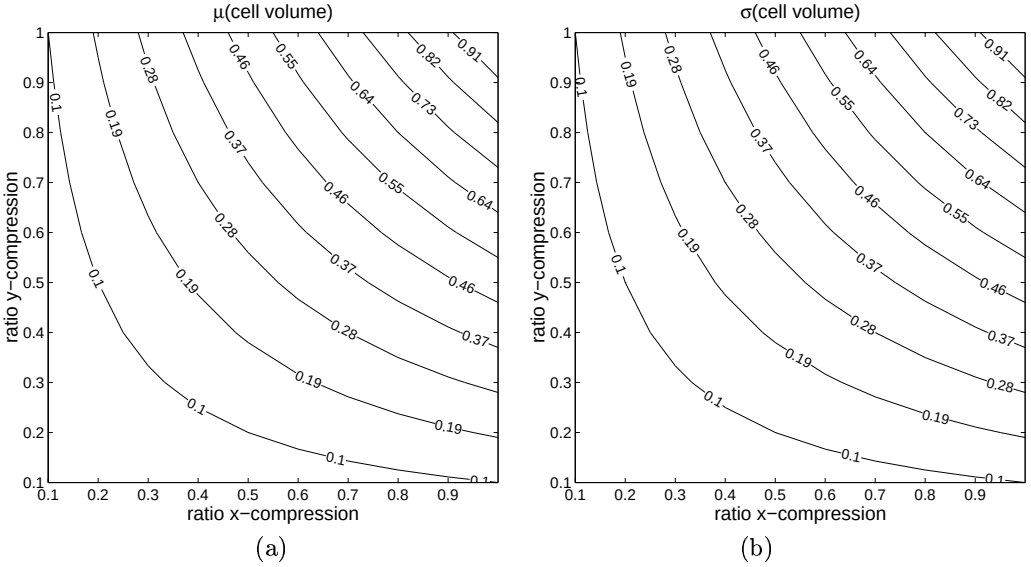



**Figure 3.53** *Volume of compressed Voronoi cells;* (a) $\mu(V_{ij}))$*and*(b) $\sigma(V_{ij})$.

Table 3.20 gives the numerical values of the mean, while Table 3.21 those of the standard deviation plotted in Figure 3.53.

$\mu(V_{ij})$

| $r_y$ \ $r_x$ | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.0** | 1.0000 | 0.9000 | 0.8000 | 0.7000 | 0.6000 | 0.5000 | 0.4000 | 0.3000 | 0.2000 | 0.1000 |
| **0.9** | 0.9000 | 0.8100 | 0.7200 | 0.6300 | 0.5400 | 0.4500 | 0.3600 | 0.2700 | 0.1800 | 0.0900 |
| **0.8** | 0.8000 | 0.7200 | 0.6400 | 0.5600 | 0.4800 | 0.4000 | 0.3200 | 0.2400 | 0.1600 | 0.0800 |
| **0.7** | 0.7000 | 0.6300 | 0.5600 | 0.4900 | 0.4200 | 0.3500 | 0.2800 | 0.2100 | 0.1400 | 0.0700 |
| **0.6** | 0.6000 | 0.5400 | 0.4800 | 0.4200 | 0.3600 | 0.3000 | 0.2400 | 0.1800 | 0.1200 | 0.0600 |
| **0.5** | 0.5000 | 0.4500 | 0.4000 | 0.3500 | 0.3000 | 0.2500 | 0.2000 | 0.1500 | 0.1000 | 0.0500 |
| **0.4** | 0.4000 | 0.3600 | 0.3200 | 0.2800 | 0.2400 | 0.2000 | 0.1600 | 0.1200 | 0.0800 | 0.0400 |
| **0.3** | 0.3000 | 0.2700 | 0.2400 | 0.2100 | 0.1800 | 0.1500 | 0.1200 | 0.0900 | 0.0600 | 0.0300 |
| **0.2** | 0.2000 | 0.1800 | 0.1600 | 0.1400 | 0.1200 | 0.1000 | 0.0800 | 0.0600 | 0.0400 | 0.0200 |
| **0.1** | 0.1000 | 0.0900 | 0.0800 | 0.0700 | 0.0600 | 0.0500 | 0.0400 | 0.0300 | 0.0200 | 0.0100 |

**Table 3.20** *Numerical values of the mean and standard deviation of compressed cell volume.*

$\sigma(V_{ij})$

| $r_y$ \ $r_x$ | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.0** | 1.0000 | 0.9000 | 0.8000 | 0.7000 | 0.6000 | 0.5000 | 0.4000 | 0.3000 | 0.2000 | 0.1000 |
| **0.9** | 0.9000 | 0.8100 | 0.7200 | 0.6300 | 0.5400 | 0.4500 | 0.3600 | 0.2700 | 0.1800 | 0.0900 |
| **0.8** | 0.8000 | 0.7200 | 0.6400 | 0.5600 | 0.4800 | 0.4000 | 0.3200 | 0.2400 | 0.1600 | 0.0800 |
| **0.7** | 0.7000 | 0.6300 | 0.5600 | 0.4900 | 0.4200 | 0.3500 | 0.2800 | 0.2100 | 0.1400 | 0.0700 |
| **0.6** | 0.6000 | 0.5400 | 0.4800 | 0.4200 | 0.3600 | 0.3000 | 0.2400 | 0.1800 | 0.1200 | 0.0600 |
| **0.5** | 0.5000 | 0.4500 | 0.4000 | 0.3500 | 0.3000 | 0.2500 | 0.2000 | 0.1500 | 0.1000 | 0.0500 |
| **0.4** | 0.4000 | 0.3600 | 0.3200 | 0.2800 | 0.2400 | 0.2000 | 0.1600 | 0.1200 | 0.0800 | 0.0400 |
| **0.3** | 0.3000 | 0.2700 | 0.2400 | 0.2100 | 0.1800 | 0.1500 | 0.1200 | 0.0900 | 0.0600 | 0.0300 |
| **0.2** | 0.2000 | 0.1800 | 0.1600 | 0.1400 | 0.1200 | 0.1000 | 0.0800 | 0.0600 | 0.0400 | 0.0200 |
| **0.1** | 0.1000 | 0.0900 | 0.0800 | 0.0700 | 0.0600 | 0.0500 | 0.0400 | 0.0300 | 0.0200 | 0.0100 |

**Table 3.21** *Numerical values of the mean and standard deviation of compressed cell volume.*

Perimeters prove to be the most difficult to find. Figure 3.54 (a) and (b) show the mean and standard deviation of the cell perimeter while Table 3.22 lists the $\mu(s_c)$ matrix that makes up one of these graphs. The values before normalisation are $\mu(s_{ij}) = 3.0158$ and $\sigma(s_{ij}) = 0.4291$.

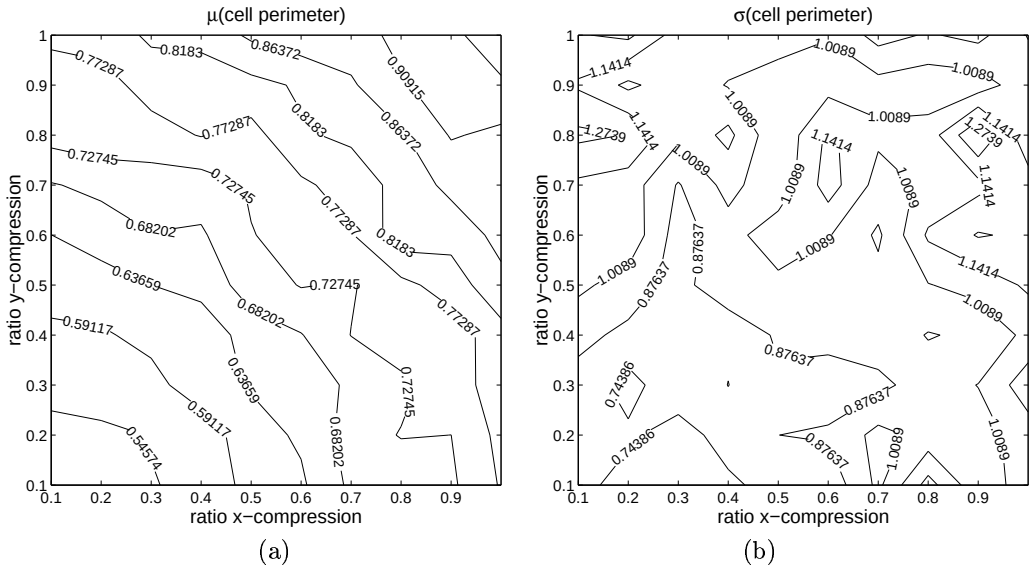



**Figure 3.54** *Cell perimeter;* (a) *the normalised mean, i.e.* $\mu(s_{ij})/\mu(s_{11})$*, and* (b) *the standard deviation,* $\sigma(s_{ij})/\sigma(s_{11})$.

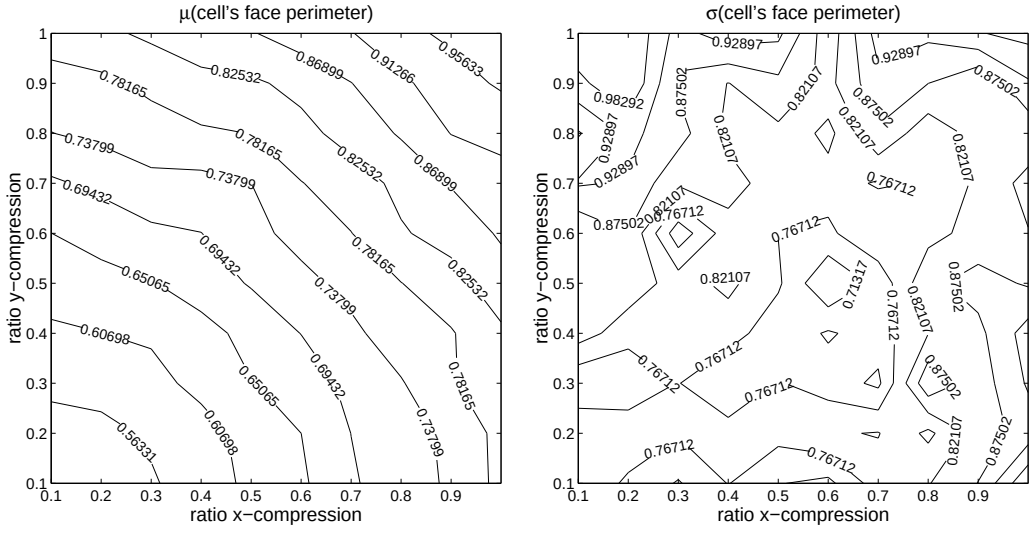
$\mu(s_{ij})$

| $r_y$ \ $r_x$ | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.0** | 1.0000 | 0.9376 | 0.9227 | 0.8878 | 0.8918 | 0.8717 | 0.8324 | 0.8271 | 0.7906 | 0.7835 |
| **0.9** | 0.9437 | 0.9333 | 0.9039 | 0.8577 | 0.8241 | 0.8046 | 0.7922 | 0.7906 | 0.7556 | 0.7472 |
| **0.8** | 0.8996 | 0.9127 | 0.8580 | 0.8266 | 0.8131 | 0.7558 | 0.7752 | 0.7560 | 0.7580 | 0.7441 |
| **0.7** | 0.8840 | 0.8766 | 0.8334 | 0.7928 | 0.7644 | 0.7313 | 0.7054 | 0.7042 | 0.6937 | 0.6788 |
| **0.6** | 0.8780 | 0.8339 | 0.8473 | 0.7684 | 0.7392 | 0.7261 | 0.6757 | 0.6806 | 0.6575 | 0.6363 |
| **0.5** | 0.8509 | 0.7948 | 0.7592 | 0.7228 | 0.7301 | 0.7017 | 0.6550 | 0.6364 | 0.6166 | 0.6036 |
| **0.4** | 0.8019 | 0.7457 | 0.7649 | 0.7284 | 0.6798 | 0.6613 | 0.6011 | 0.6036 | 0.5858 | 0.5847 |
| **0.3** | 0.7852 | 0.7609 | 0.7129 | 0.6883 | 0.6618 | 0.6408 | 0.6161 | 0.5767 | 0.5755 | 0.5662 |
| **0.2** | 0.7860 | 0.7273 | 0.7305 | 0.7008 | 0.6431 | 0.6189 | 0.5638 | 0.5476 | 0.5334 | 0.5264 |
| **0.1** | 0.7771 | 0.7198 | 0.6895 | 0.6911 | 0.6296 | 0.6090 | 0.5538 | 0.5440 | 0.5124 | 0.5003 |

**Table 3.22** *Numerical means of the cell perimeter of compressed Voronoi.*

Figure 3.55 shows the plot of the change in the perimeter of faces when the Voronoi structure is compressed. The normalised mean and standard deviation are shown in the form of contours. The values before normalisation are $\mu(\mu(z_{ij})) = 0.3766$ and $\sigma(\mu(z_{ij})) = 0.0532$.

Both the surface and the perimeter are embedded in three dimensions, the former as 2-d facets while the latter as 1-d facets. But, comparing Figure 3.54 (a) with Figure 3.54 (a), the trend of changes in the mean is smoother for surfaces than for perimeters. The difference between the threnf of the standard deviation is even more pronounced, as can be readily seen by comparing Figure 3.54 (b) with Figure 3.54 (b). This may imply the reduction in the correlation between different statistical properties as the structural differences increase, as surface is directly related to the convex hull of the volume while perimeter is a convex hull embedded in another convex hull. Also, this means that convex hull as a mapping or function is not smooth, or it could mean that it is a smooth function only up to the order one, that is to say, when we apply it only once not twice or more.




(a) (b)

**Figure 3.55** *Face perimeter;* (a) *its normalised mean, i.e.* $\mu(\mu(z_{ij}))/\mu(\mu(z_{11}))$, *and* (b) *the standard deviation,* $\sigma(\mu(s_{ij}))/\sigma(\mu(s_{11}))$.

$\mu(\mu(z_{ij}))$

| $r_y$ \ $r_x$ | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.0** | 1.0000 | 0.9640 | 0.9460 | 0.9103 | 0.8907 | 0.8659 | 0.8523 | 0.8339 | 0.8155 | 0.8056 |
| **0.9** | 0.9620 | 0.9384 | 0.9040 | 0.8681 | 0.8371 | 0.8186 | 0.8124 | 0.7927 | 0.7717 | 0.7606 |
| **0.8** | 0.9262 | 0.9137 | 0.8738 | 0.8409 | 0.8129 | 0.7823 | 0.7755 | 0.7608 | 0.7489 | 0.7372 |
| **0.7** | 0.8956 | 0.8722 | 0.8355 | 0.8058 | 0.7801 | 0.7378 | 0.7246 | 0.7276 | 0.7053 | 0.6874 |
| **0.6** | 0.8745 | 0.8404 | 0.8214 | 0.7807 | 0.7527 | 0.7252 | 0.6936 | 0.6849 | 0.6661 | 0.6504 |
| **0.5** | 0.8485 | 0.8093 | 0.7808 | 0.7514 | 0.7255 | 0.6977 | 0.6734 | 0.6479 | 0.6368 | 0.6283 |
| **0.4** | 0.8186 | 0.7780 | 0.7614 | 0.7284 | 0.6951 | 0.6659 | 0.6335 | 0.6139 | 0.6065 | 0.5987 |
| **0.3** | 0.8125 | 0.7771 | 0.7347 | 0.7075 | 0.6670 | 0.6490 | 0.6210 | 0.5916 | 0.5772 | 0.5731 |
| **0.2** | 0.7958 | 0.7459 | 0.7225 | 0.6946 | 0.6506 | 0.6254 | 0.5870 | 0.5685 | 0.5528 | 0.5466 |
| **0.1** | 0.7938 | 0.7451 | 0.7154 | 0.6897 | 0.6433 | 0.6181 | 0.5815 | 0.5594 | 0.5350 | 0.5196 |

**Table 3.23** *Numerical mean values of the face perimeter in a compressed Voronoi.*

| $\sigma(\mu(z_{ij}))$ | | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| | **1.0** | 1.0000 | 0.9795 | 0.9413 | 0.9524 | 0.7858 | 0.9426 | 0.9614 | 0.9060 | 1.0364 | 1.0311 |
| | **0.9** | 0.9221 | 0.8227 | 0.8754 | 0.9145 | 0.8047 | 0.8613 | 0.8197 | 0.8653 | 1.0369 | 0.9671 |
| | **0.8** | 0.8363 | 0.8407 | 0.7857 | 0.8670 | 0.7517 | 0.8090 | 0.8055 | 0.8980 | 0.9424 | 0.8702 |
| | **0.7** | 0.8939 | 0.8322 | 0.7718 | 0.7612 | 0.7903 | 0.7904 | 0.8453 | 0.8296 | 0.9233 | 0.9350 |
| | **0.6** | 0.8724 | 0.8413 | 0.8031 | 0.8155 | 0.7560 | 0.7662 | 0.7972 | 0.6784 | 0.8740 | 0.8279 |
| $r_y$ | **0.5** | 0.8689 | 0.8957 | 0.8640 | 0.7293 | 0.6592 | 0.7756 | 0.8340 | 0.7986 | 0.8512 | 0.8750 |
| | **0.4** | 0.9538 | 0.8608 | 0.8199 | 0.7414 | 0.7733 | 0.7324 | 0.7916 | 0.7954 | 0.7916 | 0.8463 |
| | **0.3** | 0.9359 | 0.8457 | 0.9174 | 0.7016 | 0.7422 | 0.7711 | 0.7139 | 0.7697 | 0.7137 | 0.7206 |
| | **0.2** | 0.8985 | 0.8447 | 0.7545 | 0.8238 | 0.8155 | 0.7855 | 0.7918 | 0.7724 | 0.8133 | 0.8138 |
| | **0.1** | 1.0322 | 0.8763 | 0.8181 | 0.7953 | 0.6991 | 0.7159 | 0.7692 | 0.7085 | 0.7546 | 0.8212 |

**Table 3.24** *Numerical standard deviations of the face perimeter in Voronoi network under various degrees of compression.*

The simulation above is then repeated again for a second time, where now $\mu(s_{11}) = 2.1206$, $\sigma(s_{11}) = 0.8154$, $\mu(\mu(z_{11})) = 0.3840$ and $\sigma(\mu(z_{11})) = 0.0518$. Figure 3.56 shows the results for both the cell perimeter $s$ and the face perimeter $z$. Also, compare these pictures with Table 3.25.

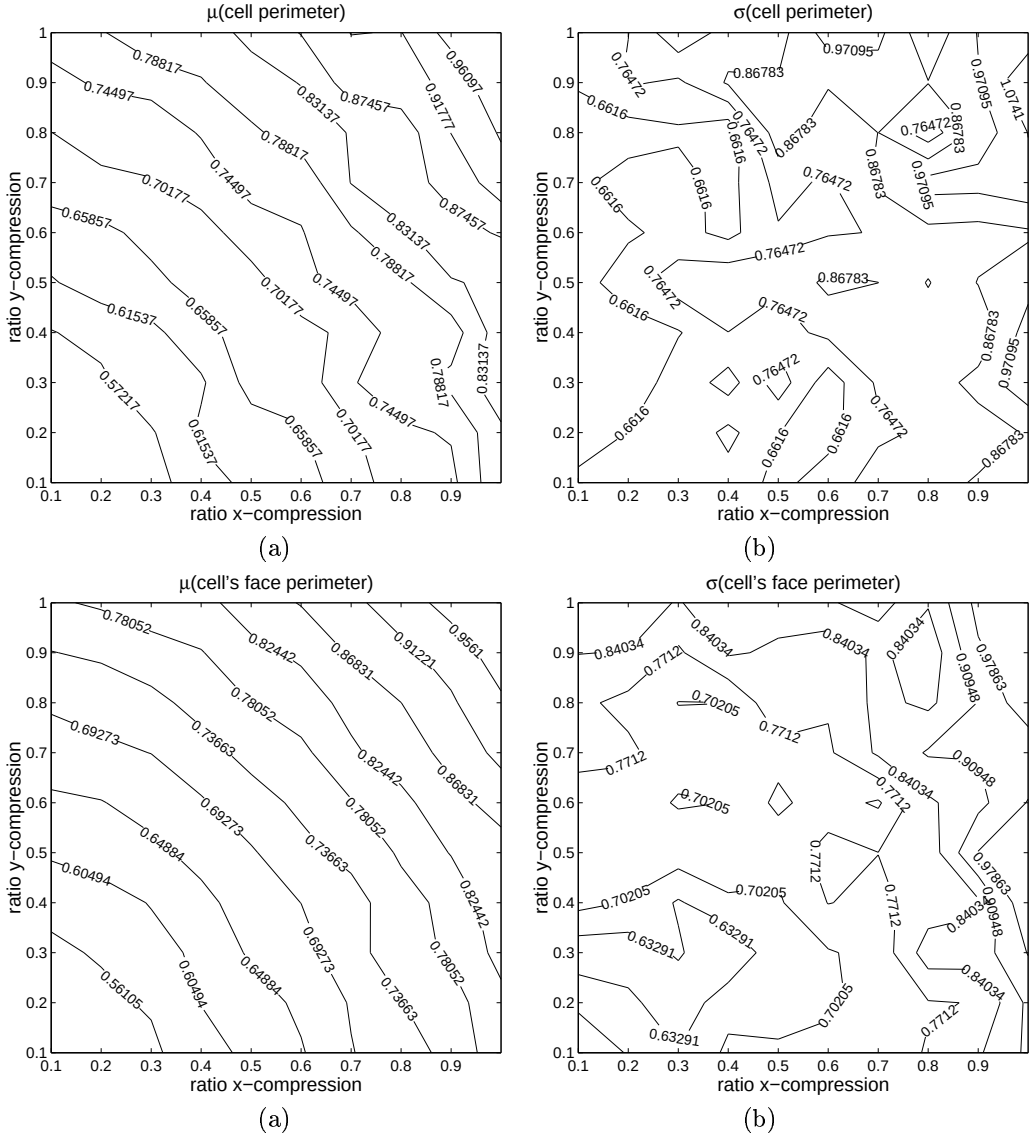



**Figure 3.56** *The cell perimeter and the face perimeter;* (a) $\mu(s_{ij})/\mu(s_{11})$, (b) $\sigma(s_{ij})/\sigma(s_{11})$, (c) $\mu(\mu(z_{ij}))/\mu(\mu(z_{11}))$ *and* (d) $\sigma(\mu(z_{ij}))/\sigma(\mu(z_{11}))$

$\boxed{\mu(s_{ij})}$

| $r_y$ \ $r_x$ | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.0** | 1.0000 | 0.9781 | 0.9164 | 0.9188 | 0.8766 | 0.8417 | 0.8126 | 0.8282 | 0.7834 | 0.7633 |
| **0.9** | 1.0042 | 0.9292 | 0.9004 | 0.8959 | 0.8330 | 0.8144 | 0.7851 | 0.7555 | 0.7511 | 0.7323 |
| **0.8** | 0.9654 | 0.9185 | 0.8512 | 0.8339 | 0.8090 | 0.7808 | 0.7475 | 0.7256 | 0.7120 | 0.7015 |
| **0.7** | 0.9422 | 0.8868 | 0.8596 | 0.8327 | 0.7596 | 0.7573 | 0.7262 | 0.6988 | 0.6964 | 0.6763 |
| **0.6** | 0.8780 | 0.8621 | 0.8241 | 0.7817 | 0.7425 | 0.7278 | 0.6810 | 0.6723 | 0.6474 | 0.6397 |
| **0.5** | 0.8411 | 0.8281 | 0.7785 | 0.7604 | 0.7350 | 0.6821 | 0.6752 | 0.6475 | 0.6352 | 0.6116 |
| **0.4** | 0.8568 | 0.7661 | 0.7653 | 0.7156 | 0.6856 | 0.6833 | 0.6423 | 0.6052 | 0.5874 | 0.5698 |
| **0.3** | 0.8605 | 0.7999 | 0.7731 | 0.7409 | 0.6744 | 0.6747 | 0.6093 | 0.6080 | 0.5627 | 0.5641 |
| **0.2** | 0.8234 | 0.7491 | 0.7368 | 0.6950 | 0.6551 | 0.6370 | 0.6231 | 0.5672 | 0.5677 | 0.5361 |
| **0.1** | 0.8255 | 0.7332 | 0.7161 | 0.6898 | 0.6345 | 0.6265 | 0.5977 | 0.5548 | 0.5552 | 0.5290 |

(a)

$\boxed{\sigma(s_{ij})}$

| $r_y$ \ $r_x$ | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.0** | 1.0000 | 0.9773 | 0.7894 | 0.9984 | 1.0195 | 0.8271 | 0.8252 | 0.9487 | 0.7626 | 0.7128 |
| **0.9** | 1.1772 | 0.9638 | 0.8714 | 0.9202 | 0.8793 | 0.8753 | 0.8798 | 0.7462 | 0.7780 | 0.6670 |
| **0.8** | 1.0766 | 0.9053 | 0.7051 | 0.8686 | 0.7943 | 0.9349 | 0.5831 | 0.6456 | 0.6093 | 0.6356 |
| **0.7** | 1.0506 | 1.0094 | 1.0137 | 0.8884 | 0.7487 | 0.7961 | 0.6260 | 0.7008 | 0.7110 | 0.6383 |
| **0.6** | 0.8534 | 0.8277 | 0.8383 | 0.7692 | 0.7554 | 0.7554 | 0.6306 | 0.6991 | 0.6439 | 0.5835 |
| **0.5** | 0.9281 | 0.8727 | 0.7584 | 0.8685 | 0.8945 | 0.7790 | 0.8540 | 0.8201 | 0.7273 | 0.6109 |
| **0.4** | 1.0266 | 0.7761 | 0.8214 | 0.7686 | 0.7893 | 0.7270 | 0.7635 | 0.6527 | 0.6022 | 0.5956 |
| **0.3** | 1.0474 | 0.9116 | 0.7985 | 0.7823 | 0.6053 | 0.8193 | 0.6155 | 0.7399 | 0.5585 | 0.5668 |
| **0.2** | 0.8807 | 0.8048 | 0.7829 | 0.7438 | 0.6276 | 0.6633 | 0.7936 | 0.6742 | 0.6638 | 0.6000 |
| **0.1** | 0.8996 | 0.8820 | 0.8161 | 0.8172 | 0.7064 | 0.6345 | 0.7206 | 0.6647 | 0.7240 | 0.6899 |

(b)

$\boxed{\mu(\mu(z_{ij}))}$

| $r_y$ \ $r_x$ | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.0** | 1.0000 | 0.9740 | 0.9327 | 0.9028 | 0.8710 | 0.8447 | 0.8119 | 0.8070 | 0.7864 | 0.7749 |
| **0.9** | 0.9817 | 0.9287 | 0.9023 | 0.8735 | 0.8320 | 0.8126 | 0.7782 | 0.7608 | 0.7442 | 0.7353 |
| **0.8** | 0.9482 | 0.9068 | 0.8681 | 0.8373 | 0.8039 | 0.7809 | 0.7508 | 0.7248 | 0.7079 | 0.7004 |
| **0.7** | 0.9246 | 0.8814 | 0.8465 | 0.8174 | 0.7699 | 0.7500 | 0.7208 | 0.6933 | 0.6843 | 0.6670 |
| **0.6** | 0.8846 | 0.8544 | 0.8126 | 0.7773 | 0.7453 | 0.7189 | 0.6878 | 0.6667 | 0.6467 | 0.6424 |
| **0.5** | 0.8510 | 0.8268 | 0.7861 | 0.7496 | 0.7240 | 0.6879 | 0.6644 | 0.6370 | 0.6243 | 0.6118 |
| **0.4** | 0.8542 | 0.7924 | 0.7657 | 0.7184 | 0.6930 | 0.6720 | 0.6354 | 0.6071 | 0.5865 | 0.5697 |
| **0.3** | 0.8378 | 0.7876 | 0.7612 | 0.7213 | 0.6755 | 0.6554 | 0.6135 | 0.5900 | 0.5643 | 0.5549 |
| **0.2** | 0.8126 | 0.7632 | 0.7354 | 0.6962 | 0.6575 | 0.6289 | 0.6062 | 0.5653 | 0.5530 | 0.5329 |
| **0.1** | 0.8088 | 0.7494 | 0.7170 | 0.6906 | 0.6449 | 0.6155 | 0.5873 | 0.5531 | 0.5413 | 0.5172 |

(c)

$\boxed{\sigma(\mu(z_{ij}))}$

| $r_y$ \ $r_x$ | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1.0** | 1.0000 | 0.9998 | 0.8469 | 0.9409 | 0.9015 | 0.8691 | 0.9077 | 0.8331 | 0.8906 | 0.8925 |
| **0.9** | 1.0290 | 0.9684 | 0.7922 | 0.8566 | 0.7914 | 0.8284 | 0.8495 | 0.7599 | 0.8379 | 0.8420 |
| **0.8** | 1.0285 | 0.9118 | 0.8258 | 0.8582 | 0.7783 | 0.8267 | 0.7019 | 0.7007 | 0.7504 | 0.7876 |
| **0.7** | 0.9241 | 0.9131 | 0.9166 | 0.8512 | 0.7612 | 0.7458 | 0.7123 | 0.7265 | 0.7861 | 0.7844 |
| **0.6** | 0.9830 | 0.8897 | 0.8278 | 0.6931 | 0.7288 | 0.7882 | 0.7028 | 0.6968 | 0.7300 | 0.7506 |
| **0.5** | 0.9891 | 0.9817 | 0.7976 | 0.7720 | 0.8001 | 0.7218 | 0.7600 | 0.7386 | 0.7329 | 0.7131 |
| **0.4** | 1.0478 | 0.8275 | 0.8022 | 0.7546 | 0.7731 | 0.6930 | 0.6875 | 0.6249 | 0.6975 | 0.7238 |
| **0.3** | 0.9820 | 0.8738 | 0.8755 | 0.7251 | 0.6961 | 0.6874 | 0.5638 | 0.6426 | 0.5881 | 0.5863 |
| **0.2** | 0.9251 | 0.7740 | 0.7667 | 0.7174 | 0.6943 | 0.6578 | 0.6800 | 0.5811 | 0.6519 | 0.6929 |
| **0.1** | 0.9230 | 0.8312 | 0.7762 | 0.7384 | 0.7160 | 0.7186 | 0.7152 | 0.6357 | 0.6987 | 0.7378 |

(d)

**Table 3.25** *Numerical values of cell and face perimeters.* (a) $\mu(s_{ij})/\mu(s_{11})$, (b) $\sigma(s_{ij})/\sigma(s_{11})$, (c) $\mu(\mu(z_{ij}))/\mu(\mu(z_{11}))$ *and* (d) $\sigma(\mu(s_{ij}))/\sigma(\mu(s_{11}))$.

Codes for finding volume, surface area, cell- and face perimeters can be found in § A.26. There only some of the printing commands have been left out.

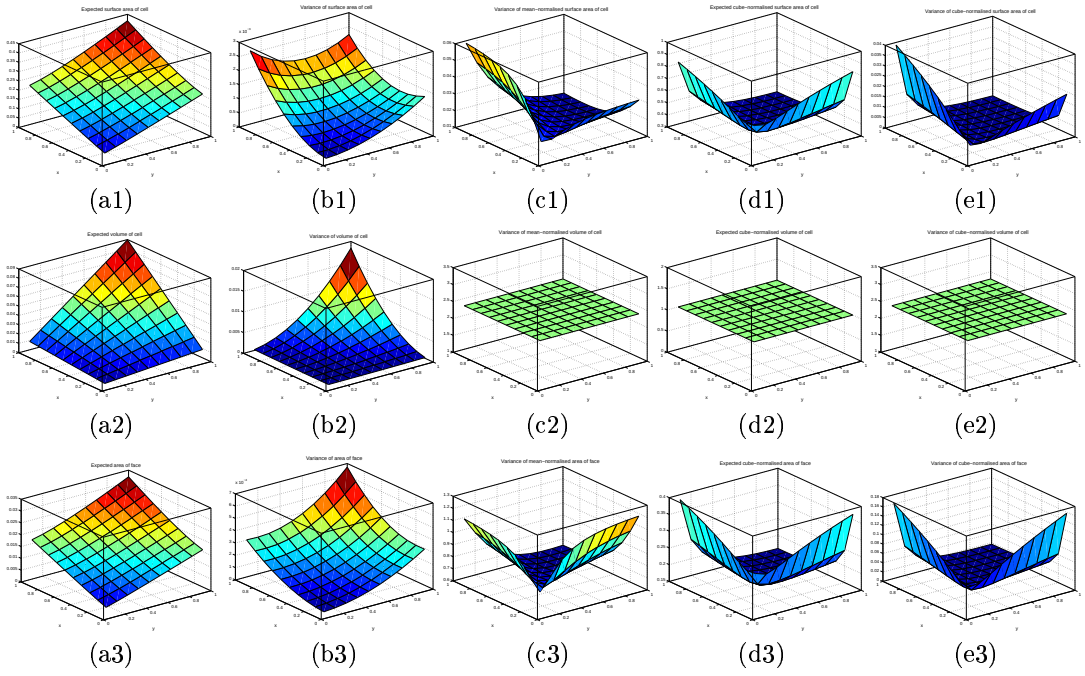
The procedure in general follows Algorithm 3.3. The volume is the summation of all tetrahedral volumes obtained from the triangulation. The surface area is summed over all triangular faces of a convex hull. Plane parameters are then calculated for all of these faces, *viz.* $a = |1, y_i, z_i|$, $b = |x_i, 1, z_i|$, $c = |x_i, y_i, 1|$ and $d = |x_i, y_i, z_i|$, in order to group them together into polyhedral faces.

**Algorithm 3.3** *Volume, area and perimeter algorithms.*

$(v, v_c) \leftarrow$ Voronoi tessellation;
**exclude** boundary vertices;
**exclude** boundary cells;
**for** every cell **do**
  **find** its Delaunay triangulation;
  $V \leftarrow \sum |x_i, y_i, z_i, 1|/6$;
  **find** its convex hull;
  $A \leftarrow \sum (s(s-a)(s-b)(s-c))^{1/2}$;
  **for** all facets of the convex hull **do**
    $(a, b, c, d) \leftarrow$ plane parameters;
    **group** hull facets into polyhedral faces;
    **for** all hull facets in every face **do**
      $\{v_f\} \leftarrow$ vertices of all coplanar facets;
      **find** convex hull of $v_f$;
      $\beta \leftarrow$ sum area of these hull segments;
      **count** number of occurrences of their edges;
      $\{e_f\} \leftarrow$ edges counted only once;
      $\{e_c\} \leftarrow e_f$;
      $z \leftarrow \sum (\sum (\Delta x_i)^2)^{1/2}$;
    **endfor**
  **endfor**
  $s \leftarrow (\sum z_i)/2$;
**endfor** □

The program `varea.m` was written long before `vareac.m`. It is listed after the latter in § A.26, although most of the variable names have been changed. The old names were long, for example `CubeNormalVolumePerInnerCell` which is now changed into `cbnafn`. The original program has not been published (Tiyapan, 2001, KNT2(ii)).

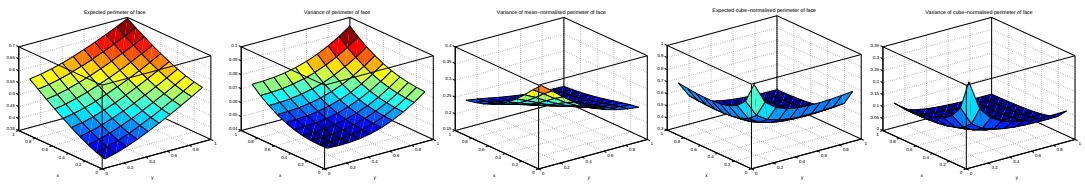
Some of the past results are shown in Figure 3.57.



(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)



(a4) (b4) (c4) (d4) (e4)

**Figure 3.57** *Past results of area, volume and perimeter.* (a), (b), (c), (d) *and* (e) *are respectively expected value, variance, variance of mean-normalised values, expected cube-normalised value and variance of the cube-normalised values, whereas* (1) *are values for surface area,* (2) *the volume of cell,* (3) *the area of face and* (4) *the perimeter of face.*

In addition, there is Table 3.26 lists the numerical statistics of the area and perimeter of face. Both this table and Figure 3.57 are from an unpublished work (Tiyapan, 2001, KNT2(ii) *ibid.*).

| $\beta_{ij}$ | | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| | **1.0** | 0.0322 | 0.0300 | 0.0278 | 0.0257 | 0.0237 | 0.0218 | 0.0200 | 0.0183 | 0.0169 | 0.0158 |
| | **0.9** | 0.0301 | 0.0280 | 0.0259 | 0.0239 | 0.0219 | 0.0201 | 0.0183 | 0.0167 | 0.0153 | 0.0142 |
| | **0.8** | 0.0280 | 0.0260 | 0.0240 | 0.0221 | 0.0202 | 0.0184 | 0.0167 | 0.0151 | 0.0138 | 0.0127 |
| | **0.7** | 0.0261 | 0.0241 | 0.0222 | 0.0203 | 0.0185 | 0.0168 | 0.0151 | 0.0136 | 0.0122 | 0.0112 |
| | **0.6** | 0.0242 | 0.0223 | 0.0204 | 0.0186 | 0.0169 | 0.0152 | 0.0136 | 0.0121 | 0.0107 | 0.0097 |
| $r_y$ | **0.5** | 0.0224 | 0.0205 | 0.0188 | 0.0170 | 0.0153 | 0.0136 | 0.0120 | 0.0106 | 0.0092 | 0.0082 |
| | **0.4** | 0.0206 | 0.0189 | 0.0172 | 0.0155 | 0.0138 | 0.0122 | 0.0106 | 0.0091 | 0.0078 | 0.0067 |
| | **0.3** | 0.0191 | 0.0174 | 0.0157 | 0.0140 | 0.0124 | 0.0108 | 0.0092 | 0.0077 | 0.0064 | 0.0052 |
| | **0.2** | 0.0177 | 0.0161 | 0.0144 | 0.0128 | 0.0112 | 0.0096 | 0.0080 | 0.0065 | 0.0051 | 0.0038 |
| | **0.1** | 0.0166 | 0.0150 | 0.0134 | 0.0118 | 0.0102 | 0.0086 | 0.0070 | 0.0054 | 0.0039 | 0.0025 |

(a)

| $\sigma(\mu(z_{ij}))$ | | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| | **1.0** | 0.6952 | 0.6728 | 0.6514 | 0.6313 | 0.6126 | 0.5956 | 0.5806 | 0.5679 | 0.5581 | 0.5516 |
| | **0.9** | 0.6733 | 0.6504 | 0.6285 | 0.6079 | 0.5886 | 0.5711 | 0.5555 | 0.5423 | 0.5320 | 0.5251 |
| | **0.8** | 0.6526 | 0.6291 | 0.6067 | 0.5855 | 0.5657 | 0.5475 | 0.5313 | 0.5175 | 0.5067 | 0.4995 |
| | **0.7** | 0.6330 | 0.6090 | 0.5861 | 0.5643 | 0.5438 | 0.5251 | 0.5082 | 0.4938 | 0.4824 | 0.4747 |
| $\delta(y)$ | **0.6** | 0.6147 | 0.5903 | 0.5668 | 0.5444 | 0.5234 | 0.5039 | 0.4865 | 0.4714 | 0.4593 | 0.4511 |
| | **0.5** | 0.5981 | 0.5731 | 0.5491 | 0.5261 | 0.5045 | 0.4844 | 0.4662 | 0.4504 | 0.4377 | 0.4289 |
| | **0.4** | 0.5832 | 0.5578 | 0.5332 | 0.5097 | 0.4875 | 0.4667 | 0.4478 | 0.4313 | 0.4178 | 0.4084 |
| | **0.3** | 0.5706 | 0.5447 | 0.5196 | 0.4956 | 0.4727 | 0.4514 | 0.4318 | 0.4145 | 0.4002 | 0.3900 |
| | **0.2** | 0.5605 | 0.5342 | 0.5087 | 0.4842 | 0.4608 | 0.4389 | 0.4186 | 0.4006 | 0.3856 | 0.3746 |
| | **0.1** | 0.5537 | 0.5270 | 0.5012 | 0.4763 | 0.4525 | 0.4301 | 0.4094 | 0.3908 | 0.3750 | 0.3633 |

(a)

**Table 3.26** (a) *The area of face and* (b) *the perimeter of face.*

These much earlier results show in addition the face area statistics. Having seen this, I thereby develop another set of codes to do the same job in the style I am using now. Programming styles change not only because I develop as a programmer, as I truely hope, but also because in mid 2001 I used a different program to create the Voronoi tessellation to what I am using now. It used to be `qhull` then, but now I am using `voronoin` on Matlab. This could explain why there are so many programs listed in the appendix, from page 249 to no less than page 342. It is not only a good practice to use more than one program for doing the same job, if only to cross check among them, but also it is my habit as a programmer. When I write a program, I usually improvise with all my facilities and experiences at that time. And as I do change in time, the programs I write tend to do likewise. This is why it is difficult to leave out any certain piece of codes. Moreover, what appear here are already merely snapshots which have survived from this continually changing virtual working ground.

The results for the face area of a VT under compressions are made up of $\mu(\mu(\beta_{ij}))$, $\sigma(\mu(\beta_{ij}))$, $\mu(\sigma(\beta_{ij}))$ and $\sigma(\sigma(\beta_{ij}))$. These are normalised respectively by $\mu(\mu(\beta_{11})) = 0.0065$, $\sigma(\mu(\beta_{11})) = 0.0018$, $\mu(\sigma(\beta_{11})) = 0.0065$ and $\sigma(\sigma(\beta_{11})) = 0.0029$, where $\beta_{ij}$ is the face area under a compression such that $r_x = i$ and $r_y = j$. These resusts are shown together as Figure 3.58.

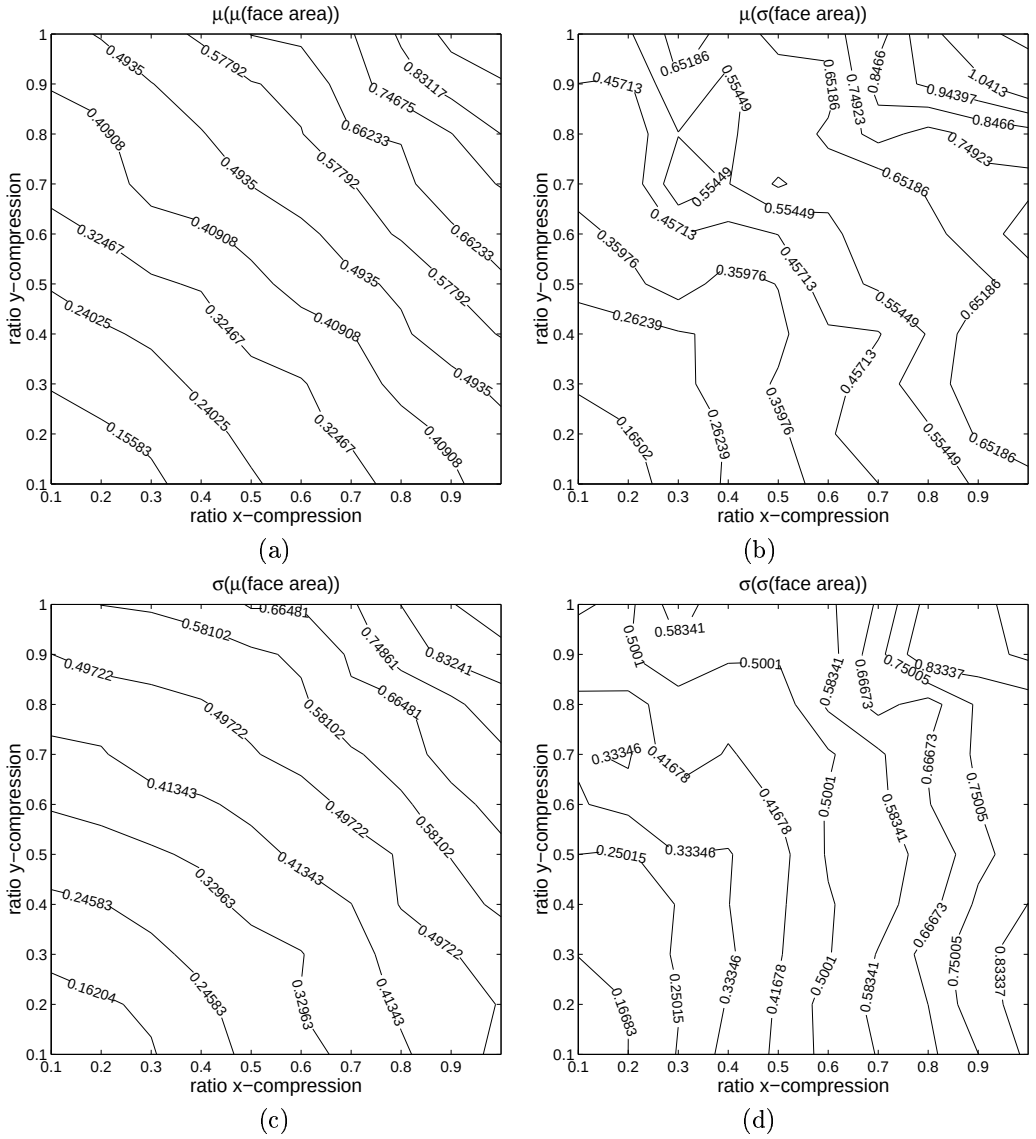



**Figure 3.58** *Statistics of face areas under compression;* (a) *the normalised mean of mean* $\mu(\mu(\beta_{ij}))/\mu(\mu(\beta_{11}))$, (b) *standard deviation of mean* $\sigma(\mu(\beta_{ij}))/\sigma(\mu(\beta_{11}))$, (c) *mean of standard deviation* $\mu(\sigma(\beta_{ij}))/\mu(\sigma(\beta_{11}))$ *and* (d) $\sigma(\sigma(\beta_{ij}))/\sigma(\sigma(\beta_{11}))$ *standard deviation of standard deviation.*

Table 3.27 has the normalised numerical values used for plotting these contours.

| $\beta_{ij}$ | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| **1.0** | 1.0000 | 0.9427 | 0.8435 | 0.7402 | 0.6837 | 0.6645 | 0.5893 | 0.5488 | 0.5002 | 0.4608 |
| **0.9** | 0.9052 | 0.8674 | 0.7970 | 0.7112 | 0.5991 | 0.5730 | 0.5306 | 0.4914 | 0.4389 | 0.4155 |
| **0.8** | 0.8316 | 0.7443 | 0.6698 | 0.6590 | 0.5745 | 0.5346 | 0.4893 | 0.4592 | 0.3924 | 0.3679 |
| **0.7** | 0.7487 | 0.7311 | 0.6343 | 0.5752 | 0.5438 | 0.4958 | 0.4651 | 0.4474 | 0.3593 | 0.3512 |
| **0.6** | 0.7221 | 0.6408 | 0.5886 | 0.5331 | 0.4699 | 0.4371 | 0.3904 | 0.3614 | 0.3302 | 0.2965 |
| **0.5** | 0.6392 | 0.5815 | 0.5073 | 0.4759 | 0.4408 | 0.3837 | 0.3311 | 0.3157 | 0.2762 | 0.2468 |
| **0.4** | 0.5825 | 0.5418 | 0.4803 | 0.4062 | 0.3724 | 0.3489 | 0.2880 | 0.2522 | 0.2293 | 0.2007 |
| **0.3** | 0.5183 | 0.4655 | 0.4324 | 0.3713 | 0.3183 | 0.2948 | 0.2527 | 0.2132 | 0.1886 | 0.1624 |
| **0.2** | 0.4633 | 0.4208 | 0.3781 | 0.3317 | 0.3141 | 0.2619 | 0.2139 | 0.1702 | 0.1422 | 0.1151 |
| **0.1** | 0.4478 | 0.3951 | 0.3437 | 0.3059 | 0.2757 | 0.2296 | 0.1870 | 0.1410 | 0.1030 | 0.0714 |

(a)

| $\beta_{ij}$ | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| **1.0** | 1.0000 | 1.0757 | 1.0285 | 0.8041 | 0.7212 | 0.6669 | 0.6257 | 0.6937 | 0.5409 | 0.5482 |
| **0.9** | 1.1387 | 1.0314 | 0.9593 | 0.8929 | 0.5949 | 0.6301 | 0.5234 | 0.6477 | 0.4421 | 0.4561 |
| **0.8** | 0.8035 | 0.7713 | 0.7159 | 0.7835 | 0.6792 | 0.5606 | 0.5531 | 0.5506 | 0.3955 | 0.4186 |
| **0.7** | 0.7238 | 0.7318 | 0.6668 | 0.5928 | 0.5840 | 0.6665 | 0.5509 | 0.6229 | 0.3910 | 0.3941 |
| **0.6** | 0.7988 | 0.6993 | 0.6248 | 0.6091 | 0.5329 | 0.4589 | 0.4261 | 0.4609 | 0.3835 | 0.3318 |
| **0.5** | 0.6963 | 0.6241 | 0.5705 | 0.5704 | 0.5172 | 0.3629 | 0.3178 | 0.4090 | 0.3335 | 0.2881 |
| **0.4** | 0.6974 | 0.7181 | 0.5614 | 0.4507 | 0.4437 | 0.3370 | 0.2802 | 0.2536 | 0.2256 | 0.2190 |
| **0.3** | 0.7246 | 0.6884 | 0.6228 | 0.5051 | 0.4242 | 0.3712 | 0.3013 | 0.2415 | 0.2232 | 0.1792 |
| **0.2** | 0.6779 | 0.6656 | 0.5349 | 0.4740 | 0.4543 | 0.3349 | 0.2668 | 0.2204 | 0.1519 | 0.1118 |
| **0.1** | 0.6377 | 0.5708 | 0.4823 | 0.4569 | 0.3863 | 0.3285 | 0.2744 | 0.1997 | 0.1332 | 0.0677 |

(a)

| $\beta_{ij}$ | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| **1.0** | 1.0000 | 0.9084 | 0.8333 | 0.7368 | 0.6714 | 0.6736 | 0.6061 | 0.5907 | 0.5819 | 0.5499 |
| **0.9** | 0.8728 | 0.8531 | 0.7962 | 0.6763 | 0.5947 | 0.5654 | 0.5499 | 0.5301 | 0.5263 | 0.4965 |
| **0.8** | 0.8034 | 0.7278 | 0.6479 | 0.6501 | 0.5653 | 0.5264 | 0.4911 | 0.4755 | 0.4539 | 0.4318 |
| **0.7** | 0.7317 | 0.7075 | 0.6189 | 0.5687 | 0.5301 | 0.4897 | 0.4648 | 0.4613 | 0.4055 | 0.4026 |
| **0.6** | 0.6995 | 0.6318 | 0.5663 | 0.5195 | 0.4533 | 0.4326 | 0.4019 | 0.3749 | 0.3567 | 0.3367 |
| **0.5** | 0.6402 | 0.5842 | 0.5038 | 0.4658 | 0.4286 | 0.3875 | 0.3421 | 0.3191 | 0.2929 | 0.2841 |
| **0.4** | 0.5954 | 0.5522 | 0.5028 | 0.4123 | 0.3740 | 0.3565 | 0.2989 | 0.2651 | 0.2437 | 0.2298 |
| **0.3** | 0.5370 | 0.4913 | 0.4437 | 0.3856 | 0.3260 | 0.2916 | 0.2620 | 0.2316 | 0.2010 | 0.1837 |
| **0.2** | 0.5022 | 0.4578 | 0.4273 | 0.3598 | 0.3321 | 0.2795 | 0.2300 | 0.1747 | 0.1520 | 0.1252 |
| **0.1** | 0.5170 | 0.4616 | 0.4006 | 0.3424 | 0.3132 | 0.2611 | 0.2164 | 0.1553 | 0.1106 | 0.0782 |

(a)

| $\beta_{ij}$ | $r_x$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1.0** | **0.9** | **0.8** | **0.7** | **0.6** | **0.5** | **0.4** | **0.3** | **0.2** | **0.1** |
| **1.0** | 1.0000 | 0.8684 | 0.8660 | 0.6771 | 0.5656 | 0.5652 | 0.5111 | 0.6327 | 0.4786 | 0.5118 |
| **0.9** | 0.9335 | 0.8894 | 0.9131 | 0.7232 | 0.5467 | 0.5089 | 0.5093 | 0.5697 | 0.4969 | 0.4564 |
| **0.8** | 0.7918 | 0.7649 | 0.6294 | 0.6942 | 0.5991 | 0.4490 | 0.4575 | 0.4608 | 0.3869 | 0.4027 |
| **0.7** | 0.7996 | 0.7652 | 0.6721 | 0.5689 | 0.4886 | 0.4718 | 0.4056 | 0.4858 | 0.3228 | 0.3420 |
| **0.6** | 0.8140 | 0.7501 | 0.6625 | 0.5617 | 0.5076 | 0.4206 | 0.3724 | 0.3551 | 0.3605 | 0.3261 |
| **0.5** | 0.7978 | 0.7256 | 0.5948 | 0.5663 | 0.5091 | 0.3878 | 0.3282 | 0.3337 | 0.2370 | 0.2504 |
| **0.4** | 0.8344 | 0.7665 | 0.6332 | 0.5491 | 0.4925 | 0.4023 | 0.3317 | 0.2549 | 0.1905 | 0.2046 |
| **0.3** | 0.8675 | 0.8142 | 0.6973 | 0.5885 | 0.4942 | 0.4097 | 0.3201 | 0.2617 | 0.1907 | 0.1702 |
| **0.2** | 0.8592 | 0.8078 | 0.6670 | 0.6029 | 0.5355 | 0.4258 | 0.3402 | 0.2465 | 0.1663 | 0.1066 |
| **0.1** | 0.8505 | 0.7491 | 0.6472 | 0.5872 | 0.5286 | 0.4294 | 0.3623 | 0.2548 | 0.1670 | 0.0835 |

**Table 3.27** *The numerical values of face area.*

## § 3.13 Voronoi tessellation in higher dimensions

In § 3.13 I mentioned the volume of a tetrahedron, which is $|x_i, y_i, z_i, 1|/6$ for which the absolute value must be taken before adding two or more together. This is what I found in existing literature. But I think that this definition is flawed because, for one thing, it does not work for the case of two dimensions. Imagine what happens if we let a unit cube to always have a unit volume. It works as shown in the following. I do not know whether this is a new discovery, but I discovered it by myself.

**Assumption 3.4.** *A unit cube has a unit volume in all dimensions d, $d \geq 2$.*

But, apart from the fact that it does not work for the case of two dimensions, the volume equation above still appeals to our commonsense, because it looks symmetrical, so I assume further.

**Assumption 3.4.** *The equation for volume of a d-simplex is $V = \text{abs}(|x_{ij}, 1|)/k$, where $x_{ij}$, $1 \leq j \leq d$, is the j:th coordinate of the $i^{\text{th}}$ point, $1 \leq i \leq d+1$, and 1 is a unit vector of $(d+1)$ dimensions.*

Under Assumption's 3.4 and 3.4, and by trial and error, I soon arive at the conclusion that the volume of a general $d$-simplex must be such that the $k$ Assumption 3.4 is $d!$. Therefore we arrive at Definition 3.1. Admittedly I tried $d(d-1)$ and $2^d - 2$ before looking at $d!$ and realised that this is the solution for $k$.

**Definition 3.4.** *The volume of a d-simplex is* $\text{abs}(|x_{ij}, 1|)/d!$, *where $x_{ij}$ is the $j^{\text{th}}$ coordinate of the $i^{\text{th}}$ vertex of the simplex.*

We can only test the appropriateness of our definition, *i.e.* Definition 3.4, against the cases of two and three dimensions, since four dimensions and above are unfamiliar grounds. But for cubes or hypercubes of two dimensions and above, the fit looks encouraging, as can be seen in Table 3.28.

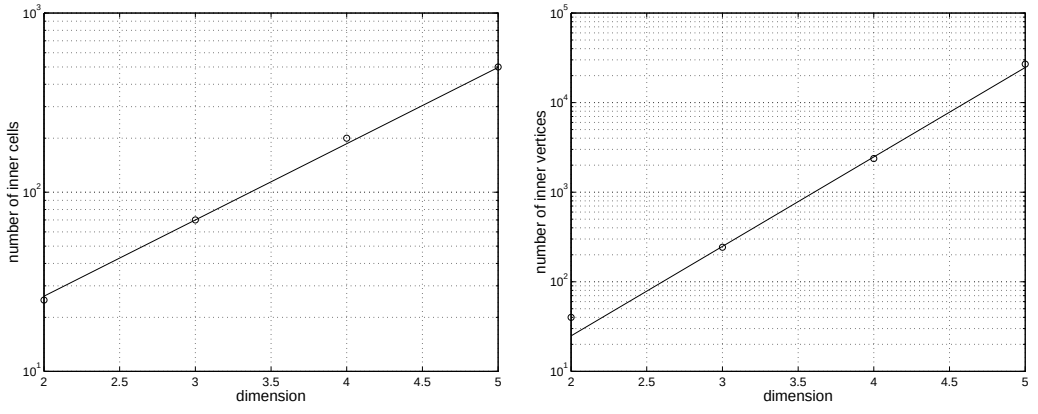
| | | $d$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| | **1** | 1 | 1 | 1 | 1 | 1 | 1 | (1) | (1) | (1) |
| $\ell$ | **2** | 4 | 8 | 16 | 32 | 64 | 128 | $(2^8 = 256)$ | $(2^9 = 512)$ | $(2^{10} = 1{,}024)$ |
| | **3** | 9 | 27 | 81 | 243 | 729 | 2,187 | $(3^8 = 6{,}561)$ | $(3^9 = 19{,}683)$ | $(3^{10} = 59{,}049)$ |

**Table 3.28** *Volume of d-hypercubes whose dimension is $\ell$. Values in brackets are implied not calculated.*

The above is, to put it in other words, our attempt at defining the equation for the volume of $d$-simplices and the test of this equation against squares, cubes and $d$-hypercubes. The equation of Definition 3.4 is also rather appealing for the reason that it allows us to calculate the volume of the cube as $V = \text{abs}(|x_{ij}, 1|)/d! = \ell^d = \prod(\Delta x_i)$, where $\ell$ is the length of the side of the hypercube and the product is over all coordinates $x_i$.

Table ℵ justifies our intuitive equation regarding the volume of hypercubes which says that such volume is $V = \ell^d$.

The program used in doing the simulations is shown in § A.27. There is also the codes I used to test the formula, which automates the generation of vertices of hypercubes before calculating their volume.

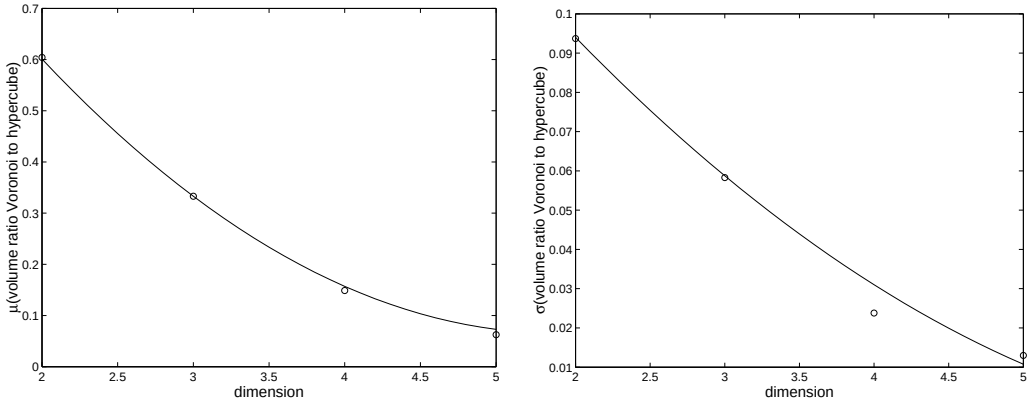



(a) (b)

**Figure 3.59** *Number of generators required for ten inner cells. The dotted line in* (a) *has the equation $y = 3.7\exp(0.98x)$ while for* (b) $y = 0.25\exp(2.3x)$.

The general procedure is to generate the Voronoi cells, exclude boundary vertices, and then boundary cells (compare Algorithm 3.3, page 120). From my experience, to produce ten usable cells, that is to say, cells in the inner reach, requires for each dimension approximately the number of generators plotted in Figure 3.59 (a). Figure 3.59 (b) is the corresponding number of vertices.

For a Voronoi cell in $d$ dimensions we may draw a box around them such that the walls of the box are all parallel to the coordinate planes. For the plane, a Voronoi graph consumes some 60 per cent such binding rectangle. For higher dimensions, this ratio goes down from 0.6 to 0.33, 0.15 and 0.6 respectively for 3, 4 and 5 dimensions. The trend for the standard deviations is also similar to this. Both the mean and the standard deviation decrease with dimension, but their curves seemed difficult to define if we try to plot it on a semilog graph. The curve turns out to be parabola, as shown in Figure 3.60.




(a) (b)

**Figure 3.60** *Mean and standard deviation of the ratio Voronoi per defining cube. Both graphs fit well with the parabolic equation* $y = ax^2 + bx + c$; (a) *has* $a = 0.046$, $b = -0.498$ *and* $c = 1.413$, *while* (b) *has* $a = 0.00375$, $b = -.054$ *and* $c = 0.187$.

Additional results on a 6-dimensional VT generated from 1,200 generators are $\mu(V_p/V_h) = 0.0252$ and $\sigma(V_p/V_h) = 8.5 \times 10^{-4}$. The results are shown in Table 3.29.

| $n_g$ | $c_{in}$ | $d$ | $\mu(V_p/V_h)$ | $\sigma(V_p/V_h)$ | $V_{in}$ |
|---|---|---|---|---|---|
| 25 | 10 | 2 | 0.6041 | 0.0937 | 33 |
| 30 | 16 | 2 | 0.6034 | 0.0769 | 44 |
| 70 | 13 | 3 | 0.3331 | 0.0583 | 241 |
| 200 | 11 | 4 | 0.1488 | 0.0238 | 2,374 |
| 500 | 9 | 5 | 0.0638 | 0.0098 | |
| 500 | 11 | 5 | 0.0625 | 0.0130 | 26,893 |
| 1,200 | 4 | 6 | 0.0252 | $8.5 \times 10^{-4}$ | 306,710 |

**Table 3.29** *Voronoi volume in higher dimensions;* $V_p$ *is the Volume of the polytope,* $V_h$ *that of the hypercube and* $n_g$ *number of the generators.*

The accumpanying curve plotted in Figure 3.61 is a parabola $y = 0.0354x^2 - 0.428x + 1.32$. On the other hand, neither the mean value nor the standard deviation of the volume ratio can possibly follow a parabolic curve, for the obvious reason that both decrease monotonically towards zero.

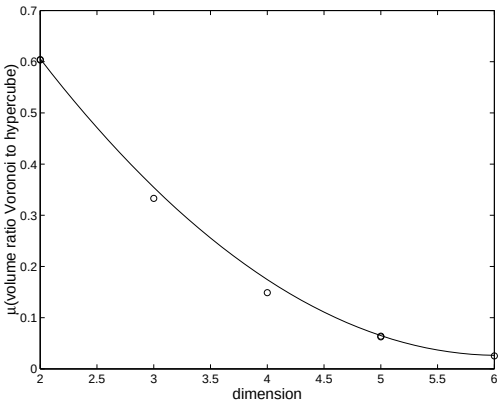



**Figure 3.61** The ratio $V_p/V_h$, dimension from 1 to 6.

## § 4. Percolation

The 1970's and the 1980's saw a proliferation of variations on the theme of percolation. Every year there seemed to be a new percolation problem or two. For a *Bethe lattice* Chalupa *et al* (1979) reported a *bootstrap percolation* where those randomly occupied sites with less than $m$ occupied neighbours are recursively emptied one by one until a stable configuration is reached. The problem they are interested in is that where the impurity concentration, dilution and crystal-field interaction compete in magnetic materials compete against the exchange interaction, resulting in the magnetic moments and consequently the magnetic order being destroyed.

Wilkinson and Willemsen (1983) introduced *invasion percolation.* Working with Schlumberger they were interested in the real problem in the oil industry where water displaces oil via capillary action. Their approach was that of a constant flow rate, not one of a constant pressure as usual. Here water displaces oil in the smallest available pores. But when water completely surrounds any region of oil, no further advance into that region will be possible, oil being incompressible. Such regions are called *trappings* and cause a problem generally known by the name of *residual oil*, an economic bane for oil industry.

In the above example the hydrophobic versus hydrophilic propertyplays an important role in the replacement of oil in pores with water. And water is prevented from penetrating trapped oil regions by much the same principle as that which prevents the water in the contents of a sandwich from crossing the spreaded layer of butter to wet the hydrophilic bread. For this latter case the pores in question are those within the bread texture, and the soaking of the bread is best prevented provided that the trapped regions of butter or margarine percolates in two dimensions to form a single layer or film which entirely covers the sectioning surface of the bread. Moreover, there is a similarity also in the internal structure of both the rock and the bread. The density of pores in bread is not homogeneous as a result of the tension on the surface of the dough caused by internal air pressure originated from the yeasts inside, as well as because of the heat applied to it when inside the oven, which dehydrolises the surface and makes it dry and hardened. The same inhomogeneity can be found inside the rocks which form the oil reservoirs where the regions of oil are surrounded by rocks which are less porous and have less permeability.

For Adler and Aharony (1988) a random walker, aka ant, treads on clusters. The ant enlarges a cluster by stepping on to an empty site next to it which meets certain conditions. They called this problem *diffusion percolation.* An example of a condition met is where the empty site has two or more occupied neighbours on a square lattice.

Most percolation in studies happens by randomly toggling the phases of sites or bonds in regular lattices. Kerstein (1983) considered randomly located spheres, take the complementary region of their union, and then perform percolation on the former. He showed that such percolation problem is equivalent to a percolation on a Voronoi lattice whose sites are the sphere centres.

In the same way as an infinite loop in computer science means that one can not come to the termination of a program going along the time dimension, an infinite cluster in percolation means that one can not come to the end of a cluster shifting along either one of the dimensions. The former case is along the one dimension of time and is only possible because of the time flows in one direction. Therefore half the line spanned is considered as being infinite. In one nondirectional dimension, except for the trivial case where one can consider the entire space as being one single cluster, no infinite cluster is possible. In simulation, when a cluster spans the whole of the space considered along any dimension we say that it is infinite since as one moves a cross section along that dimension, it always contains a section of the cluster.

The bond percolation program of two-dimensional Voronoi network by Tiyapan (1995, KNT2(ii), p. 78) takes $\mathrm{O}(n^2)$ time provided we assume that the contribution made by the number of clusters together with the that by each cluster comparison amount to a linear time complexity term. In order to justify this assumption, consider first the number of clusters. The maximum number of clusters possible depends on the size of the system, in other words it must vary as $n_1^k$, where $0 < k_1 < 1$. This maximum value should be approximately 0.5 because of symmetry between occupansy and void clusters. Next consider the time involved in comparing two clusters. On Matlab this is a sparse vector comparison which is likely to involves some linked lists, and similarly should take time as a function of $n_2^k$, $0 < k_2 < 1$. Because on average the size of clusters is always small before Pc, $k_2$ will be less than 0.5. Therefore $n_1^k \cdot n_2^k < n$ and it is safe to assume $\mathrm{O}(n)$ time from both of them combined. *q.e.d.*. A C translation (Tiyapan 1995, *ibid.*, p. 80) of this program, though not as bad as it may seem because constant coefficients are small, gives $O(n^5)$ time in comparison.

In the field of geology Miller *et al* (2000) studies the analogy between the dilatant slip in

earthquakes and the hydrofraction occurred in melting and dehydration, the percolation of the latter in the permeability network internal to the fault being the cause of the former. According to them, the existence of toggle switches in the permeability rules out the assumption that the permeability throughout the whole system is homogeneous. Instead, the system reorganises itself into sytems of different scales of interaction according to the degree and nature of its inhomogeneity. At the critical state the scale of interaction is equal to the scale of the model.

The percolation probabilities given by Stauffer and Aharony (1994) are, the first number being for sites and the second for bonds, for the honey comb lattice 0.6962, 0.65271; square 0.592746, 0.50000; triangular 0.50000, 0.34729; diamond 0.43, 0.388; simple cubic 0.3116, 0.2488; body-centred cubic 0.246, 0.1803; face-centred cubic 0.198, 0.119; 4-d hypercubic 0.197, 0.1601; 5-d hypercubic 0.141, 0.1182; 6-d hypercubic 0.107, 0.0942; 7-d hypercubic 0.089, 0.0787.

De Gennes *et al* (1959) investigated disordered binary solid solution AB where active atoms A are randomly replace the nodes of the periodic matrix B. There exists a critical concentration A in B below which all clusters are finite, and above which both finite and non-finite, *i.e.* infinite, clusters exist. Such solid solution in networks can represent the spin waves in alloys with one ferromagnetic component or the impurity bands in semiconductors. They cited seminal work on percolation by Broadbent *et al* (1957), but no mention was made about the Ising model.

In a way, percolation is similar to diffusion. In diffusion the particles considered move about randomly, whereas in percolation they can only crop up randomly at predetermined locations on a network which is fixed. We could imagine, for instance, cars running along the roads within a traffic network as diffusing through them. Then the percolation could occur on a larger scale, that is the scale of a road. The cars move along, that means they diffuse; but the roads remain fixed, and so their phases could percolate. In other words, in diffusion the particles move while in percolation, whether there are moving particles or not, it is the phases that percolate. Since historically percolation began as the study of diffusion of particles in a network of tubes in which the phases are naturally defined as the tubes being blocked or unblocked, these definitions have become most frequently used in other areas of application, for example in filtering membranes and traffic networks.

But this is not necessarily the case. Instead of dealing with a fixed network, one may consider a model of percolation in a continuum, for example by randomly patching an area until all the patches connect with one another somehow and percolate. The patch could be of any shape, as well as polygonal and circular. We can consider the percolation in a certain area as having occurred when there appears a cluster of patches which traverses any two opposite sides. One application of this is in the study of occurences of epidemics. Hoyle and Wickramasinghe (1979), having given a convincing argument in favour of viruses and various forms of diseases being carried to Earth from space by comets, talk about patchiness of pathogenic clouds. According to them, simultaneous attacks across vast region rules out person-to-person theory. Moreover, influenza epidemics are generally characterised by sudden onsets and equally sudden ends. These epidemics and plagues may be thought of as the percolation of these patches in a sufficiently large, predefined area. The bacteria and viruses coming from space adding to gene give the possibility of jump patterns in evolution (*cf* Smith and Szathmáry, 1995). The cells deliberately refuse to block viruses because they could prove to be useful in the long run, generally not by individuals but by the species. Historical examples are the disease described by Thucydides between 431 and 404 BC, five epidemics of 'English Sweats' between 1485 and 1552, and more than ten influenza pandemic from 1700 to 1900. Example of diseases which are caused by bacteria and viruses are bubonic plague, chicken pox (varicella), cholera, common cold, Legionnaires' disease, leprosy, measles, mumps, poliomyelitis, small pox, tuberculosis, and trachoma. Examples of major evolutionary transitions are those going from RNA to DNA, from prokaryotes to eukaryotes, and from axexual cloning to sexual propagation. Another example is the transition from primate to human both of whom differ from each other neurologically in the ability to use language and the power to conceptualise. One description of the Great Plague in London (Dickens, 1851). There was a rumour that a few people died in the winter of 1664. In May 1665 the disease burst out in St. Giles's which raged through July and September in every part of England; approximately 10,000 people died in London alone. But then the equinox winds virtually blew the disease away, and the Plague quickly disappeared. The existence of interstellar organic matters is supported by strong evidences with more and more complex substances constantly found (*cf* Hoyle and Wickramasinghe, 1978).

Gas turns into liquid by the growth of larger and larger clusters. But unlike percolation in networks, clusters in condensation process are not well defined. Crystalisation is also characterised by the growth of one phase within another. But here the orientations are an inherent part of the

clusters themselves, and there is a long range coordination among clusters. One important thing in percolation is for the system size to be infinite. Random discs overlapping one another in a continuum helps correct counts of bacteria cultures (*cf* Essam, 1980). Let $p_c$ be the critical probability and $P(p)$ the percolation probability. Then as $p$ approaches $p_c$ from above, $P(p) \approx (p - p_c)^{\beta}$ where $0.4 < \beta < 0.5$ is the critical exponent. If $p$ approaches $p_c$ from below, $S(p) \approx (p_c - p)^{-\gamma}$ and $\xi(p) \approx (p_c - p)^{-\nu}$ where $S(p)$ is the conditional average $\langle s|F\rangle$, $s$ the number of particles in a cluster, and $F$ the event that the origin $O$ is occupied and belongs to a finite cluster. In other words, $S(p)$ is the mean size of cluster at the origin given that $F$ occurs. This last event occurs with probability $p_F = p(1 - P(p))$ and $p_F = p$ for $p < p_c$. Furthermore, $S(p) = p_F^{-1} \sum_r C(r,p)$ and $\xi^2(p) = [p_F S(p)]^{-1} \sum_r r^2 C(r,p)$ where the pair-connectedness function is $C(r,p) = \langle \eta(r)|F\rangle p_F$, and $\eta(r)$ is the indicator defined to be one if $r$ is connected to $O$ and zero otherwise. When $F$ occurs, $s = \sum_r \eta(r)$. There are two different definitions of the critical probability, $p_c = \sup\{p|P(p) = 0\}$ and $\pi_c = \sup\{p|P(p) = 0 \text{and} S(p) < \infty\}$, sometimes denoted by $p_H$ and $p_T$ for Hammersley and Temperley respectively. By definition, $\pi_c \leq p_c$. When $p$ approaches $p_c$ from above, $\gamma'$ and $\nu'$ are similarly defined respectively by $S(p) \approx (p - p_c)^{\gamma'}$ and $\xi(p) \approx (p - p_c)^{\nu'}$ It is generally assumed as $\gamma' = \gamma$ and $\nu' = \nu$. No proofs for these assumptions exist, eventhough they are consistent with the series expansions. Estimates of $\gamma$ and $\nu$ are $1.6 < \gamma < 1.7$ and $0.8 < \nu < 0.9$. In a dilute ferromagnet a cluster containing $s$ spins each of which has a unit magnetic moment has a probability $p = \exp(\frac{1}{2}sh)/\left[\exp(-\frac{1}{2}sh) + \exp(\frac{1}{2}sh)\right]$ of being parallel to the magnetic field $H > 0$. Here $h = 2H/k_BT$ where $k_BT$ is the thermal energy. For an infinite cluster, $p = 1$. The zero-field magnetic moment is $\mu_0(p) \propto P(p)$. The zero-field magnetic moment is $\mu_0(p) \propto P(p) \sim (p - p_c)^{\beta}$. The field dependent magnetic moment at $p_c$ is $\langle (1 - \exp(-sh))/(1 + \exp(-sh))\rangle_F$ and $\frac{1}{2}(1 - G(p_c,h)) \leq \mu_c(h) \leq 1 - G(p_c,h)$ where $G(p,h) = \langle \exp(-sh)\rangle_F$. Therefore the critical probability $\delta$ is defined by $\mu_c(h) \sim 1 - G(p_c,h)h^{1/\delta}$. The correlation function between $\sigma$ on site $i$ and $j$ is defined by $\Gamma_{ij} = \langle \sigma_i\sigma_j\rangle_T - \langle \sigma_i\rangle_T\langle \sigma_j\rangle_T$ where $\langle\cdot\rangle_T$ is an average over spin states. Then $\Gamma_{ij} = 1$ when $i$ and $j$ belong to the same cluster and zero otherwise. The fluctuation formula is $k_BT\langle\chi\rangle = \sum_j \langle \Gamma_{ij}\rangle$ where $\langle \Gamma_{ij}\rangle = C(r_j - r_i, p)$, the mean susceptibility is $\langle\chi\rangle = p_F S(p)/k_BT \sim (p_c - p)^{-\gamma}$ and the mean free energy is $\langle F\rangle = (k_BT \ln 2)K(p)$ where $K(p)$ is the mean number of clusters and $K(p) \sim (p_c - p)^{2-\alpha}$, where the index assigned to the third derivative divergence is $1 + \alpha$, $\alpha \approx -0.5$.

The growth mechanism of clusters in percolation is all so found in biology. Williams and Bjerknes (1972) simulate a tumour in the basal layer of an epithelium. The basal cells become less sensitive to Charlone which controls cellular division, and thus they divide $\kappa$ times faster than the normal cells where $\kappa > 1$ is the carcinogenic advantage. Abnormal cells interior to the basal layer divide, push, and then replace the neighbouring cells leaving the overall configuration unchanged except at the border where the abnormal cells exert a thrust of $\kappa - 1$ on their normal neighbours, that is $\frac{\mathrm{d}N}{\mathrm{d}t} = (\kappa - 1)n$ where $n$ and $N$ are respectively the numbers of peripheral and total abnormal cells. They found that dimensionality of fractal is involved and the dimension 1.1, instead of 1, must be assigned to the periphery, which means that $n$ is proportional to $N^{0.55}$ not $N^{0.5}$. They found that abnormal cells push out faster in this order: the triangular, square, and hexagonal lattice. We may explain this, by looking at the coordination numbers of these three lattices, that the higher coordinaition number the lattice sites have, the slower the cluster expands. Added coordination means a higher degree-of-freedom the newly divided cells have to move about while still remaining local.

Percolation is a field full of far more open questions than discovered answers. This is one of the reasons it is more suitable to research than university examinations (*cf* Stauffer and Aharony, 1998). The fascination not so well hidden within the field is reflected in the number of research monographs written on this field that is close to percolating, which makes it impossible to give a full list of references. Percolation deals with the clusters formed by randomly occupying the each site of a very large lattice with probability $p$ independent of its neighbours. Most of the studies to date concentrate on the critical phenomena, which exist in a very narrow range in each problem, and the scaling theory which tries to describe them. The contiguity criteria are easier to define in applications in physics and physical sciences, because here problems can be described by, or easily simplified into some definite geometry, whereas in other applications this may not be the case. For example, in economic modelling the nature of the geometry or even the number of dimensions of such structure is still not well understood (*cf* Tiyapan, 1997, KNT2(ii) and KNT2(ii)), let alone the problem of contiguousness. Traffic network (*cf* Tiyapan, 1997, KNT2(ii) and KNT2(ii)) is another example. Here the flow is along channels within tubes, *i.e.* roads, both of which could be directed, which make the network a pool of tangled threads not only difficult to imagine but also to define such things as

congestion and contiguousness. Even in geographical problems like that of forest fire, the spread of fire may be defined in at least three ways, namely by next-nearest neighbour, neighbour or double neighbour respectively in cases of a common corner, a common side or two neighbours. Reservoirs occur when the petroleum, formed in sedimentary rocks, migrates into permeable sedimentary rocks like sandstone. Around 70% of North Sea oil fields also date from the Jurassic period. These include Beatrice, Brent, Cormorant, Murchison and Tartan. Oil companies want to exploit reservoirs where $p > p_c$. Moreover, they want to tap into the largest cluster in these reservoirs. For this purpose bore hole samplings are carried out which collected within $L \times L$ frames and then the number of points, $m(L)$, belonging to the same cluster counted. It turns out that $m(L)$ is proportional to $L^{1.9}$. In general $m \propto L^d$ where $d$ is either integral, in which case it is the Euclidean dimension, or nonintegral, in which case it is the fractal dimension introduced by Benoit Mandelbrot. There exists a correlation length $\xi$ such that $m(L) \propto L^{1.9}$ for $L < \xi$, and $m(L) \propto L^2$ for $L > \xi$. This correlation length is a measure of the largest hole of the largest cluster and decreases as $p$ is increased above $p_c$.

One way of finding the percolation threshold is by using the *ant in a labyrinth* algorithm shown here as Algorithm 4.1.

**Algorithm 4.1** *Ant in a labyrinth. At* $p_c$, $k = 1/3$, *whereas* $k = 0$ *for a constant distance, and* $k = 1/2$ *for normal diffusion.*

```
for various p do
  repeat a large amount of times
    occupy the sites with probability p;
    sum ← 0;
    for a large number of simulations n do
      for various t upto a large number do
        sumr ← 0;
        for i = 1 to t
          ant randomly occupies an occupied site;
          identify neighbours;
          move into an occupied neighbour randomly chosen;
          r ← distance traveled;
          sumr ← sumr + r;
        endfor
        sumrsq ← sumr^2;
      endfor
    endfor
    sum ← sum + sumrsq;
    R ← sum^(1/2);
    plot R against t in double logarithmic scales;
  endrepeat
  k ← the slope of the straight line just plotted;
endfor
p_c ← the p s.t. k = 1/3;)
```

□

Also, the number of steps the ant takes, $t$, for the linear size of a region it visited, $R$, is fractal, that is $t \propto R^{1/k}$. They give several percolation probabilities for selected lattices, which are shown in Table 4.1, as well as the exact values of $p_c$'s, which are for the square bond percolation $1/2$, triangular site $1/2$, triangular bond $2\sin(\pi/18)$, and honeycomb bond $1 - 2\sin(\pi/18)$. For the honeycomb site percolation, $p_c < 1/\sqrt{2}$. Site percolation on hypercubic lattices of high dimensions have $p_c = 1/(2d - 1)$.

| **lattice** | **site** | **bond** |
|---|---|---|
| honeycomb | 0.6962 | 0.65271 |
| square | 0.592746 | 0.50000 |
| triangular | 0.500000 | 0.34729 |
| diamond | 0.43 | 0.388 |
| simple cubic | 0.3116 | 0.2488 |
| body-centred cubic | 0.246 | 0.1803 |
| face-centred cubic | 0.198 | 0.119 |
| 4-d hypercubic | 0.197 | 0.1601 |
| 5-d hypercubic | 0.141 | 0.1182 |
| 6-d hypercubic | 0.107 | 0.0942 |
| 7-d hypercubic | 0.089 | 0.0787 |

**Table 4.1** *Percolation thresholds from Stauffer and Aharony (1998)*

There is no percolation in one dimension because in such case the percolating cluster must necessarily contain the whole space. But there are some applications where the critical blockage is important, for example the blockage of drainage grilles by pea shingle is one-dimensional. Here the critical amount of blockage depends on the critical rate of flow which in turn depends on the amount of water and the rate of accumulation of water to be drained. The maintenance of gullies and grilles is done by cleaning, flushing and grit bucket emptying (*cf* Harrison and Trotman, 2002).

For the Bethe lattice $p_v = p_e = 1/(z-1)$. In general, $p_e \leq p_v$. For a network of $d$ dimensions, $zp_e \approx d/(d-1)$. In his review, Sahimi (1994) gave some of the current know $p_c$'s which are listed in Table 4.2 here.

| | $d$ | $z$ | $p_e$ | $zp_e$ | $p_v$ |
|---|---|---|---|---|---|
| honeycomb | 2 | 3 | $1-\sin(\pi/18)$ | 1.96 | 0.6962 |
| square | 2 | 4 | 1/2 | 2 | 0.5927 |
| kagome | 2 | 4 | 0.522 | 2.088 | 0.652 |
| trianglular | 2 | 6 | $2\sin(\pi/18)$ | 2.084 | 1/2 |
| diamond | 3 | 4 | 0.3886 | 1.55 | 0.4299 |
| simple cubic | 3 | 6 | 0.2488 | 1.49 | 0.3116 |
| body-centred cubic | 3 | 8 | 0.1795 | 1.44 | 0.2464 |
| face-centred cubic | 3 | 12 | 0.198 | 1.43 | 0.119 |

**Table 4.2** *Percolation probabilities, cf Sahimi (1994).*

The accessible fraction, $f_\alpha$, is the fraction of occupied bonds that belong to the infinite cluster. The backbone fraction, $f_\beta$, is the fraction of those accessible bonds which belong to a transport path, *i.e.* a path with all dead ends excluded. The correlation length, $\lambda$, is the length scale over which the random network is macroscopically homogeneous. In a Monte Carlo simulation it is necessary that the size $L$ of the network is sufficiently larger than this correlation length in order to obtain a $p_c$ which is independent of $L$. The expected cluster size $m$ is $\mathrm{E}(m) = \sum_m m^2 n_m / (\sum_m m n_m)$, where $n_m(p)$ is the expected number of clusters of size $m$ per lattice site and $mn_m$ the probability that a site belongs to an $m$-cluster. The fraction of the network which can accommodate flow has various names associated to each application, for example the effective electrical conductivity $g_e$, the effective diffusivity $D_e$ and the hydrodynamic permeability $k$. The effective elastic moduli, $G$, are the elastic moduli of the network a fraction $p$ of bonds of which are elastic elements, while the rest are rigid or stiff. The fraction of isolated occupied sites is $f_i(p) = p - f_\alpha(p)$. The universal scaling laws state that $p_c(p) \sim (p-p_c)^{b_p}$, $f_\alpha(p) \sim (p-p_c)^{b_p}$, $f_\beta(p) \sim (p-p_c)^{b_\beta}$, $\lambda(p) \sim |p-p_c|^{-\nu_p}$, $k(p) \sim |p-p_c|^{-\gamma_p}$, $g_e(p) \sim (p-p_c)^\mu$ and $G(p) \sim (p-p_c)^f$, where $b_\beta$, $b_p$, $\nu_p$ and $\gamma_p$ are topological exponents, completely universal, depend only on the dimensionality but not the microscopic details of the system. The effective diffusivity $D_e(p) \sim (p-p_c)^{\mu-b_p}$ because $g_e \sim n_e D_e$ and $n_e \sim f_\alpha(p)$. Near $p_c$, $k(p) \sim (p-p_c)^e$. For network percolations $e = \mu$, but for the continuum percolation this may not be the case.

As $p \to p_c^-$, perfectly conductive clusters become larger and $g_e$ increases. Then $g_e(p) \sim (p_c - p)^{-s}$ near $p_c$ and diverges at $p_c$. In two dimensions, $\mu = m$. If $p$ edges are totally rigid while the rest are elastic, then $G$ diverges as $p \to p_c^-$ such that $G \sim (p_c - p)^{-\zeta}$ For large $m$ near $p_c$, $n_m \sim m^{-\tau_p} f\left[(p-p_c)m^{\sigma_p}\right]$, where $\tau_p$ and $\sigma_p$ are universal and $f(0) \neq 0$. Some relations among the geometrical exponents are $\tau_p = 2 + b_p\sigma_p$ and $\nu_p d = b_p + 1/\sigma_p = 2b_p + \gamma_p$. Sahimi (*ibid.*) listed some values of the current critical exponents and fractal dimensions which we list again here as Table 4.3.

| | $d=2$ | $d=3$ | Bethe lattices |
|---|---|---|---|
| $b_p$ | 5/26 | 0.41 | 1 |
| $b_\beta$ | 0.48 | 1.05 | 2 |
| $\nu_p$ | 4/3 | 0.88 | 1/2 |
| $\gamma_p$ | 43/18 | 1.82 | 1 |
| $\sigma_p$ | 36/91 | 0.45 | 1/2 |
| $\tau_p$ | 187/91 | 2.18 | 5/2 |
| $D_c$ | 91/48 | 2.52 | 4 |
| $D_\beta$ | 1.64 | 1.8 | 2 |
| $D_{\min}$ | 1.13 | 1.34 | 2 |
| $\mu$ | 1.3 | 2.0 | 3 |
| $m$ | 1.3 | 0.73 | 0 |

**Table 4.3** *Critical exponents and fractal dimensions, cf Sahimi (1994).*

The system is macroscopically homogeneous when $L \gg \lambda$ but not when $L \ll \lambda$ where the cluster spanning the sample is self-similar and fractal at all length scales up to $\lambda$. Its mass is $M \sim \lambda^{D_c}$, where $D_c = d - b_p/\nu_p$ is the fractal dimension of the cluster. For $L \gg \lambda$, $D_c = d$. Similarly the backbone is also a fractal object when $L \ll \lambda$, and it has the fractal dimension $D_\beta = d - b_\beta/\nu_p$. When $L < \lambda$ one should replace $\lambda$ by $L$, at $\lambda$ diverges at $p = p_c$ and the red bonds have $M_r \sim L^{D_r}$, $D_r = 1/\nu_p$, when $L \ll \lambda$. The minimal or chemical path between two points of a percolation cluster is the shortest path between the two points, and $L_{\min} \sim L^{D_{\min}}$ for $L \ll \lambda$.

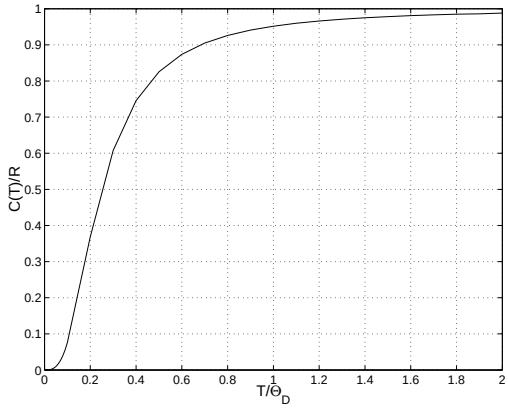
Finite-size scaling is the effect of finite size on the critical properties. As $p \to p_c$, $\lambda \to L$, and the variations of a property $P_L$ becomes $P_L \sim L^{-x} f(u)$, where $u = L^{1/\nu_p}(p - p_c) \sim (L/\lambda)^{1/\nu_p}$ and $f(0) \neq 0$. When $p \to p_c$ and $L \to \infty$, $P_\infty \sim (p - p_c)^\delta$ and $x = \delta/\nu_p$. There is also a shift in the percolation threshold, and $p_c - p_c(L) \sim L^{-1/\nu_p}$ where the percolation probability is $p_c$ for the infinite system while its effective value for a finite system is $p_c(L)$. Correction should be made whenever the system simulated is small, that is $P_L \sim L^{-x}\left[a_1 + a_2 g_1(L) + a_2 g_2(L)\right]$, where $g_1$ and $g_2$ are the correction terms to the scaling. For transport properties, for example conductivity, diffusivity and elastic moduli, $g_1 = (\ln L)^{-1}$ and $g_2 = L^{-1}$ are often good enough.

Sahimi (1994) mentions a random distribution of inclusions, such as circles, spheres, or ellipses, in an otherwise uniform system or continua; a percolation on random polyhedra such as the VT; and also a random distribution of random conductin sticks of with given aspect ratio, or plates a given extent, which is important in modelling fracture networks in rocks. In the study of percolation in continua, one considers a continuous, random function $h(r)$, defined for all points $r$'s in the entire space, such that $\langle h(r) \rangle = 0$. All regions of space where $h(r) < R$ are in a phase which is different from the rest, the volume of which goes from zero to infinite as $R$ changes from negative- to positive infinite. As $R$ becomes larger, so do the islands, and the phase grows and percolates at the critical $R_c$. For continua percolation, the critical occupied volume fraction is defined as $\phi_c = p_c f$, where $f$ is the filling factor of a lattice when all its sites are occupied by an impermeable sphere in such a manner that those which are nearest neighbours touch each other. This volume fraction seems to be an invariant, approximately 0.45 for two- and 0.15–0.17 for three dimensions. If the spheres are permeable, then $\phi_c = 1 - \exp(-B_c/8)$ where $B_c = z p_b$ is the number of bonds per sites at $p_c$, and in continua percolation $B_c \to p_c z$ as $z \to \infty$. If $\phi_c$ is an invariant of continuous systems, then $R_c$ is given by $\phi_c = \int_{-\infty}^{R_c} h(r) \mathrm{d}V$, where $V$ is the volume of the system.

## § 4.1 S-curves and the percolative phenomena

Before any abrupt change occurs there must be a graduation process leading up to it. In processes where the time constant is long, the characteristic s-shape is obvious, take for example a learning curve. But even where time constants are short, I doubt if such thing as a strictly abrupt change exists. At the very point of transition, undoubtedly there may be a singularity, for instance at the Big Bang. But even the Big Bang can not exist alone by itself. To preserve the symmetry of nature, there must be another process at the other end leading up to it. It is only because such process must necessarily be on the other side from us and we can not see it from here. That side may belong to the antimatter or the anti-universe, but one thing I believe, whenever there is a singularity there must be a symmetry.

My past work in § E.1 has been include here to be used as an example. The per cent extraction curves which I drew then (Tiyapan, 1991) all show the change to be abrupt, starting off from zero time immediately with a positive gradient. This can not happen in the real world, so there must have been a foot of the characteristic s-curve at the beginning. I must have been that my sampling time is too long, and anyhow the nature of the reaction may make it impossible to observe the development in detail with accuracy.




The predicted value of the heat capacity of monoatomic solids according to the Debye theory, from de Podesta (1996), is shown in Figure 4.1. According to Debye, $C_V = (1944p/\Theta_D^3)T^3$ JK$^{-1}$mol$^{-1}$ (*cf* de Podesta, 1996). The curve in Figure 4.1 shows the characteristic s-shape which should have appeared in my past research publication described in § E.1.

**Figure 4.1** predicted value of the heat capacity of monoatomic solids according to Debye.

The foot of a positive s-curve has a positive, nonzero second derivative. This corresponds with the positive cooperativity of the product curve in studies of enzyme assay and kinetics. In enzyme assay, the product versus time graph shows a positive cooperativity characteristic when the Hill constant $h > 1$ in the equation $y_s = [s]^h/(K + [s]^h)$, where $y_s$ is the fractional saturation of the enzyme with substrate while $s$ the concentration of the substrate (Eisenthal and Danson, 2002). Cooperativity reflects the equilibium binding of substrate or other ligand. The binding of a substrate molecule to an enzyme either facilitates, when the cooperativity is positive, or hinders, when the same is negative, the binding among molecules of the same substrate.

## § 4.2 Voronoi percolation in two dimensions

The percolation of Voronoi tessellation in two dimensions can be achieved by the following algorithm.

```
generate random points;
generate Voronoi tessellation and Delaunay triangulation;
find vertices within the square box bounded by lower and upper bounds;
find neighbours of all cells, bonds, vertices, and edges;
for unit = cell, bond, vertice and edge do
  find number of units;
  permute list of units;
  clear cluster list 1, cluster list 2, set of resultant clusters;
  cluster percolated ← false;
  for i = 1 to number of units do
    existing cluster joined = false;
    for j = 1 to number of clusters in cluster list 1 do
      if the j^th cluster contains the i^th unit in permuted list do
        merge the i^th unit into the j^th cluster ;
        existing cluster joined ← true;
      end
      if existing cluster joined is true
        move the j^th cluster of cluster list 1 to cluster list 2 ;
        for k = 1 to number of clusters in cluster list 1 do
          if the k^th cluster in cluster list 1 touches the cluster in cluster list 2 do
            merge the former into the latter;
          else
            append the former to cluster list 2;
          end
        end
        test percolation of the cluster just updated;
        move cluster list 2 to cluster list 1;
        clear cluster list 2;
        break;
      end
    end
    if existing cluster joined is false
      create a new cluster of size one and append it to cluster list 1;
    end
    append cluster list 1 to set of resultant clusters ;
  end
end
```

**Algorithm 4.2** *Network percolation in 2-d.*

Pc probabilities are found to be ${}^{Vn\ 2d}_{4(1),14(2)}p_c = 0.5110 \pm 0.0856$, ${}^{Vn\ 2d}_{17(2)}p_b = 0.3095 \pm 0.0523$, ${}^{Vn\ 2d}_{16(2)}p_v = 0.7231 \pm 0.0616$ and ${}^{Vn\ 2d}_{16(2)}p_e = 0.6801 \pm 0.0468$.

And coordination numbers are $.2436\ {}^{Vn\ 2d}_{2(1),7(2)}x_c = 5.2320$ , $.2979\ {}^{Vn\ 2d}_{8(2)}x_b = 9.5022$ , $.0259\ {}^{Vn\ 2d}_{8(2)}x_v = 2.8617$ and $.0382\ {}^{Vn\ 2d}_{8(2)}x_e = 3.8064$ .

The percolation theory in two dimensions has benefitted much from discoveries regarding the Ising model in Physics. The Ising model is to date probably the most successful model in percolation theory. This is not only because it can describe in details the phenomena of ferromagnetism and antiferromagnetism, but also because it does so by replacing a noncrystalline solid in the atomic scale with a perfect lattice.

When a bar of iron is placed in a external magnetic field at a constant temperature, the field will induce some magnetisation in the bar. If the external field is slowly turned off, there are two scenarios possible. If $T < T_c$ the bar retains some of its internal magnetisation, but if $T > T_c$ the magnetisation completely disappears. The transition temperature $T_c$ is known as the Curie point.

The Ising model represents the iron bar by lattices, for instance a square lattice. Then it defines two power series, namely the low- and the high temperature series. The low temperature series comes from a bivariate generating function which is generated from the number of colourings for which there are $p$ black sites and $q$ black-white edges. In other words if we let $A(p, q)$ be the said

number of colourings, then our bivariate generating function becomes $a(x,y) = \sum_{p,q} A(p,q)x^p y^q$ and the power series is obtained from $\alpha(x,y)\lim_{N\to\infty}(1/N)\ln(a(x,x))$.

On the other hand, the high temperature series of the Ising model changes its form with the dimension. In one dimension it degenerates, while in two dimensions it is isomorphic to the low temperature case and comes from a bivariate generating function that is generated from the number of even polygonal drawings whose area is $p$ and which has $q$ edges. In three dimensions it comes from a univariate function that in turn is generated by the number of even polygonal drawings which have $q$ edges. An even polygonal drawing on a lattice is a union of its subgraphs that uses each edge of the latter at most once and each site an even number of times. It is also known as an Eulerian subgraph and is indeed a union of simple, closed and edge-disjoint polygons which need not be connected.

In three dimensions if we let $B(q)$ be the number of even polygonal drawings mentioned, then our univariate generating function becomes $b(z) = \sum_q B(q)z^q$ and the power series can then be obtained from $\beta(z) = \lim_{N\to\infty}(1/N)\ln(b(z))$.

Two main problems remain unsolved regarding the Ising model, namely that of finding closed-form expressions, of $\alpha(x,y)$ for two dimensions and of $\beta(z)$ for three.

## § 4.3 Voronoi percolation in three dimensions

In three dimensions the procedure of finding $p_c$ is similar to that used in the 2-d case. Algorithm 4.3 is what I wrote and used in doing the simulation. I developed it to run on Matlab, and therefore do every thing in matrix form. As a consequence, there are various types of data all of which are matrices. These data structures can be summarised into the following.

A *list* is a one-dimensional matrix whose dimension is the number of its members. An *index* is a matrix whose members are either one or zero. These index matrices in two dimensions are like a map or an array. They are normally one or two dimensions, and map the relationship between the members of one set and another. A *set* is an $m \times n$ matrix every one of the members $a_{ij}$ of which is a matrix. A *pair matrix* is a square diagonal index matrix that maps among members within the same set. It contains the information about relationships like neighbourhood, group membership, connection, *etc.* A pair matrix or an index can also contain information other than the existence or membership flags, for example the edge length in the case of a vertices pair edge matrix. In a bond or an edge the mid point represents its position.

The program can be divided into three parts, creating and arranging the Voronoi data, finding the neighbourhood matrix and the percolation simulation for $p_c$. Putting the vertices of a face in order amounts to juggling from one end of each stick, *i.e.* edge, to another. Edges are traced in one direction only until all the edges are successfully linked head to tail. Two lists receive the result from the tracing, vertices which match go to one of them while those which do not is put in the other and recycled.

**Algorithm 4.3** *Managing the Voronoi data in three dimensions.*

$x \leftarrow$ **create** random points;
$(v_a, c_a) \leftarrow$ **create** Voronoi tessellation;
$t \leftarrow$ **create** Delaunay tessellation;
**find** $v_n$ such that for all $v_a$, $0 < v_a < 1$;
**find** $c_n$ such that all vertices of $c_a$ are in $v_n$;
$c \leftarrow c_n$;
**for** all **do**
  $m_b \leftarrow$ **find** mid bond coordinates;
  $b_l \leftarrow$ **find** bond length;
**endfor**
**for** all pairs of cells **do**
  $f_a \leftarrow$ **find** shared vertices when either of the two cells is in $c_n$;
**endfor**
**for** all $v_n$, $v \leftarrow v_a$;
**for** all $c_n$, $f \leftarrow f_a$;
**for** all $f$ **do**
  **if** it has three vertices **then**
    all the three possible pair combinations are neighbours;
  **else**

```
    (a, b, c) ← find the face equation from three vertices;
    θ_i ← ka where k = 1/(a² + b² + c²) and i = 1, 2 and 3;
    max ← 0;
    for j = 1 to 3 do
      if θ < max then
        j_m ← j;
        max ← θ(j_m);
      endif
    endfor
    if j_m = 1 then
      p ← y; q ← z;
    elseif j_m = 2 then
      p ← x; q ← z;
    else
      p ← x; q ← y;
    endif
    d ← find delaunay triangulation from p and q;
    for all edges of all triangles of d do
      record the number of times they occur;
    endfor
    neighbours ← vertices of edges which occur only once;
    for all f do
      order all their vertices;
    end
  endif
endfor
```

□

The terms *vertices* and *edges* refer to the Voronoi tessellation only. The vertices and edges of the Delaunay tessellation are cells and bonds of the VT. The program `perco3d.m` can find $p_c$ for all of these. For each one of them we must find the neighbourhood matrix as well as lists of the upper- and the lower boundaries. The order of blockages is completely decided in advance by finding a permuted list of numbered items. The program finds the history of clusters for the whole range of $p$, *i.e.* from 0 to 1.

To obtain the matrix of cell neighbours, first crop out between 5 and 10 per cent the boundary in all directions. Then the neighbour matrix is found from the DT matrix.

The bond neighbour matrix is simply the rearrangement of the cell neighbour matrix obtained. We have already found the vertices neighbour matrix earlier while searching for vertices shared between cells. This is rearranged for the use in the percolation program, and then again for the edge neighbour matrix.

Next is Algorithm 4.4 which carries out the percolation simulation. The cluster information is swapped alternately between three variables, $A$, $B$ and $tmp$, to facilitate the flow; $A(i)$ is the $i^{\text{th}}$ cluster in $A$. All variables are assumed to be cleared at the start of the algorithm. At the end of each run the value of $p_c$ is computed, and the list $p$ is reversed and reused for a second time if it was the first run.

**Algorithm 4.4** *Voronoi percolation in three dimensions.*

```
for each permuted item p(i) in the list do
  joined ← 0;
  for each cluster A(j) in A do
    if A(j) contains p(i) then
      A(j) ← A(j) ∩ p(i);
      B(1) ← A(j);
      tmp ← A;
      A ← tmp − A(j);
      for all A(k) in A do
        if A(k) ∧ B(1) then
          B(1) ← A(k) ∩ B(1);
        else
          B(++n_B) ← A(k);
```

```
        endif
      endfor
      percolated ?;
      A ← B;
      break;
    endif
  endfor
  if joined then
    A(++n_A) ← p(i);
  endif
endfor
```

□

The resulted Pc probabilities from simulation are ${}^{Vn\ 3d}_{12(2)}p_c = 0.2340 \pm 0.0448$, ${}^{Vn\ 3d}_{2(2),10(3)}p_b = 0.1178 \pm 0.0271$, ${}^{Vn\ 3d}_{2(2),10(3)}p_v = 0.2941 \pm 0.0831$ and ${}^{Vn\ 3d}_{12(3)}p_e = 0.4311 \pm 0.0324$.

And the coordination numbers are $.3329\ {}^{Vn\ 3d}_{6(2)}x_c = 11.1231$, $.5996\ {}^{Vn\ 3d}_{1(2),5(3)}x_b = 23.0548$, $.0245\ {}^{Vn\ 3d}_{1(2),5(3)}x_v = 3.6926$ and $.0432\ {}^{Vn\ 3d}_{6(3)}x_e = 5.5002$.

Four different types of percolation have been performed, namely cell, bond, vertice and edge percolation. Vertices are inherently zero dimension, edges and bonds one dimension and cells, which in reality have three dimensions, is considered for this purpose as having none. When considering vertice- and cell percolation, the position of the vertices and nuclei determines whether they could connect somewhere to somewhere else. When the object the percolation of which we consider has a nonzero dimension and the dimension of the network is small, that is to say, in the same order as that of itself, $p_c$ obtained will depend on the network dimension. In other words, there will be an influence from the size of the objects on, and distort, the space in which the percolation occurs.

Additional results on a larger network are, the Pc probabilities from simulation are ${}^{Vn\ 3d}_{20(2)}p_c = 0.2039 \pm 0.0410$, ${}^{Vn\ 3d}_{20(3)}p_b = 0.0963 \pm 0.0114$, ${}^{Vn\ 3d}_{20(3)}p_v = 0.1659 \pm 0.0571$ and ${}^{Vn\ 3d}_{28(3)}p_e = 0.4172 \pm 0.0242$, and the coordination numbers are ${}^{Vn\ 3d}_{1(3)}x_c = 12.4177$, ${}^{Vn\ 3d}_{,1(3)}x_b = 25.7077$, ${}^{Vn\ 3d}_{1(3)}x_v = 3.8126$ and ${}^{Vn\ 3d}_{1(3)}x_e = 5.7050$. In this case the network was originally built from 1,000 Poisson point generators, and for the simulation we have $n_c = 723$, $n_b = 4{,}489$, $n_v = 4{,}855$ and $n_e = 9{,}255$.

## § 4.4 Percolation of 2-dimensional Voronoi sections

The program in § A.4 does a 2-d section of the 3-d VT. Algorithm 4.5 describes how it works. It assumes that the Voronoi tessellation already exists. Here $d_m$ means a denominator while $n_r$ a numerator, $V$ and $C$ means vertices and cells of the 3-d Voronoi tessellation while $v$ and $c$ those of the 2-d section. In particular, $c \in C$

**Algorithm 4.5** *Plane section of Voronoi in three dimensions*

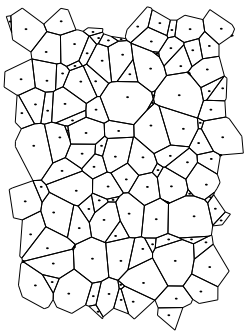
```
(Δx, Δy, Δz)_i ← (x_2 − x_1, y_2 − y_1, z_2 − z_1)_i for all edges i;
for all edges i do
  d_m ← aΔx + bΔy + cΔz;
  if d_m nonzero then
    n_r ← ax_1 + by_1 + cz_1;
    t ← −n_r/d_m;
    if 0 ≤ t ≤ 1 then
      (x, y, z) ← (x_1 + Δx, y_1 + Δy, z_1 + Δz)_i;
      {v_s} ← (x, y, z);
    endif
  else
    a ← a + ε;
    d_m ← aΔx + bΔy + cΔz;
    n_r ← ax_1 + by_1 + cz_1;
    t ← −n_r/d_m;
    if 0 ≤ t ≤ 1 then
      (x, y, z) ← (x_1 + Δx, y_1 + Δy, z_1 + Δz)_i;
      {v_s} ← (x, y, z);
    endif
    if n_r = 0 then
```

$\{v_s\} \leftarrow (x_1, y_1, z_1)$;
$\{v_s\} \leftarrow (x_2, y_2, z_2)$;
**endif**
**endif**
**endfor**
**for** all $c_s$ **do**
**find** the Delaunay triangulation;
**count** $n_e$ of all the triangles, add the numbers into a single list;
**endfor** □

The intersection of a plane $ax + by + cz + d = 0$ by the line which is defined by $x = x_1 + \Delta xt$, $y = y_1 + \Delta yt$ and $z = z_1 + \Delta zt$, where $\Delta x$, $\Delta y$ and $\Delta z$ are respectively $(x_2 - x_1)$, $(y_2 - y_1)$ and $(z_2 - z_1)$, is determined by $t = -(ax_1 + by_1 + cz_1 + d)/(ai\Delta x + b\Delta y + c\Delta z)$. If the denominator is zero the line is parallel to the plane, and if the nominator is also zero contained therein.

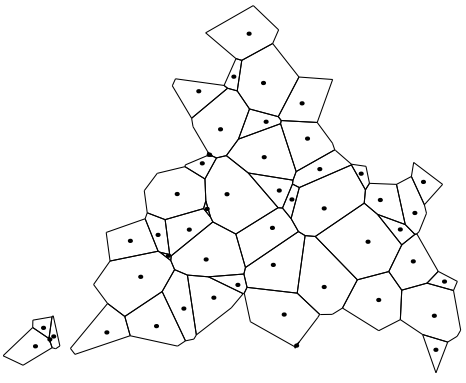
The results from the percolation simulation on the section $\mathcal{V}_3^2$ are ${}^{Vn\ (2,3)s}_{18(\mathbf{1}),4(\mathbf{2})}p_c = 0.5494 \pm 0.1223$, ${}^{Vn\ (2,3)s}_{10(\mathbf{1}),10(\mathbf{2})}p_b = 0.3515 \pm 0.0764$, ${}^{Vn\ (2,3)s}_{8(\mathbf{1}),12(\mathbf{2})}p_v = 0.7557 \pm 0.0757$, ${}^{Vn\ (2,3)s}_{20(\mathbf{2})}p_e = 0.6210 \pm 0.0665$.

The coordination numbers are $.2212\ {}^{Vn\ (2,3)s}_{8(\mathbf{1}),1(\mathbf{2})}x_c = 4.6894$, $.6021\ {}^{Vn\ (2,3)s}_{5(\mathbf{1}),4(\mathbf{2})}x_b = 8.8100$, $.0387\ {}^{Vn\ (2,3)s}_{4(\mathbf{1}),5(\mathbf{2})}x_v = 2.7495$, $.1118\ {}^{Vn\ (2,3)s}_{9(\mathbf{2})}x_e = 3.6691$.



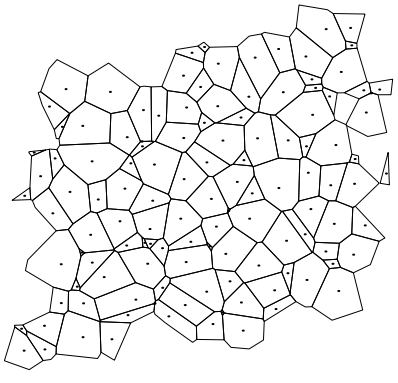
When sectioned by the plane $(a, b, c, d) = (0.01,\ 0.5,\ 0.5,\ -0.5)$, our VT gives a picture as shown in Figure 4.2. Here the points shown are merely the average of the coordinates of all vertices in each cell. From ten simulations of these 118 cells, 300 bonds, 288 vertices and 405 edges having respectively the coordinate numbers of 5.0847, 9.7933, 2.8125 and 3.7333, we obtain $p_c = 0.5220 \pm 0.0966$, $p_b = 0.3107 \pm 0.0391$, $p_v = 0.7014 \pm 0.0573$ and $p_e = 0.6259 \pm 0.0603$.

**Figure 4.2** Section by the plane (0.01, 0.5, 0.5, -0.5).



When sectioned by the plane $(a, b, c, d) = (0.5,\ -0.5,\ 0.5,\ 0.01)$, our VT gives a picture as shown in Figure 4.3. Here the points shown are merely the average of the coordinates of all vertices in each cell. From fourteen simulations of these 50 cells, 104 bonds, 140 vertices and 187 edges having respectively the coordinate numbers of 4.1600, 8.2500, 2.6714 and 3.5080, we obtain $p_c = 0.6914 \pm 0.1424$, $p_b = 0.4533 \pm 0.1108$, $p_v = 0.7760 \pm 0.0821$ and $p_e = 0.6929 \pm 0.0897$.

**Figure 4.3** Section by the plane (0.5, −0.5, 0.5, 0.01).



When sectioned by the plane $(a, b, c, d) = (-0.7,\ -0.3,\ 1,\ 0.01)$, our VT gives a picture as shown in Figure 4.4. Here the points shown are merely the average of the coordinates of all vertices in each cell. From twenty simulations of these 121 cells, 305 bonds, 291 vertices and 411 edges having respectively the coordinate numbers of 5.0413, 9.4426, 2.8247 and 3.7518, we obtain $p_c = 0.5413 \pm 0.1107$, $p_b = 0.3584 \pm 0.0494$, $p_v = 0.7077 \pm 0.0572$ and $p_e = 0.6749 \pm 0.0470$.

**Figure 4.4** Section by the plane (−0.7, −0.3, 1, 0.01).

For the purpose of these simulations, I have turned the code for finding percolation into a function. To my surprise and delight, this same function works for both the 3–d VT and its 2–d section. In choosing a sectioning plane it is better if we choose the parameter $d$ small, as the plane will then pass close to the origin. Also, choosing $a + b + c \approx 0$ seems to make a more wholesome section than otherwise. The codes for sectioning work well for oblique planes but do not like planes which are parallel to an axis. This shortcoming can be avoided if we make our plane only nearly parallel, when we want it to be parallel to an axis. Then to be able to view in a head on fashion such planes which have been plotted in three dimensions, we can look from the position $(a, b, c)$, which is in effect the vector normal to the plane. The function mentioned above is listed as `perc.m` and is in § A.24.

## § 4.5 Network percolation

In the study of networks an important parameter is the coordination number, which is the number of neighbours of an element which in the graph theory is usually the vertex. Each vertex or site of a graph is connected to each of its neighbouring vertices by a bond, so the coordination number of a graph is the number of bonds connected to a vertex.

Clusters and their various characteristics play an important role in the study of percolation of networks. With a material science application in mind, Levy *et al* (1982) numerically represent the shape of a cluster by the *shape parameter* $S$, defined as $S = b/N$ or more generally $S = (1/2)\sum_{i=1}^{z} i\nu_i / \sum_{i=1}^{z} \nu_i$ where $b$ or $i$ is the number of bonds and $N$ or $\nu_i$ the number of elements in a cluster having $b$ or $i$ bonds respectively.

## § 4.6 Percolation statistics in literature

Rushbrooke and Morgan (1961) studied the Heisenberg and Ising ferromagnetics and found that the critical concentration $p_c$ is the same for both Heisenberg and Ising problems. Moreover, this value is irrelevant to the magnitude of the spin $s$. By studying point-clusters and link-diagrams, they found approximations to the exact values of the percolation probabilities for, the face-centred cubic lattice $p_c = 0.18$, the body-centred cubic lattice $p_c = 0.22$, tthe simple cubic lattice $p_c = 0.28$, and the plane square $p_c = 0.48$. Only vertices of their point-clusters, not link-diagrams, need to have a bond between them in order to be neighbours.

Frisch *et al* (1961) give the site critical probabilities for various types of lattices in two- and three dimensions. In the terminology used here, these are $p_c$ of vertices of the networks considered. Their results are shown in Table 4.4.

| **lattice** | **dimension** | **z** | $p_c$ (N=1,000) | $p_c$ (N=2,000) |
|---|---|---|---|---|
| triangular | 2 | 6 | $0.487 \pm 0.021$ | $0.493 \pm 0.018$ |
| square | 2 | 4 | $0.575 \pm 0.017$ | $0.581 \pm 0.015$ |
| hexagonal | 2 | 3 | $0.683 \pm 0.020$ | $0.688 \pm 0.017$ |
| h.c.p. | 3 | 12 | $0.204 \pm 0.008$ | – |
| f.c.c. | 3 | 12 | $0.199 \pm 0.008$ | – |
| simple cubic | 3 | 6 | $0.325 \pm 0.023$ | – |
| tetrahedral | 3 | 4 | $0.434 \pm 0.013$ | $0.436 \pm 0.012$ |
| ice (quartz) | 3 | 4 | $0.432 \pm 0.013$ | $0.433 \pm 0.011$ |

**Table 4.4** *Critical probabilities from Frisch et al (1961).*

From their investigations they also realise that the critical probabilities of two homohedral tilings may not be the same. For instance, a plane lattice with $z = 3$ may have the vertex- and edge percolation probabilities different from those of the hexagonal lattice, eventhough the coordination number is the same for both.

Both Monte Carlo technique and series expansion method are means by which one can numerically study percolation problems. Dean (1963) used Monte Carlo method because, according to him, it provides more precise information. In a network of $N$ sites $pN$ of which are occupied, the probability of site occupation at the next time step becomes $q = p + 1/N$. The critical probabilities he found are shown in Table 4.5.

| | **lattice size** | | | |
|---|---|---|---|---|
| | $12 \times 12$ | $24 \times 24$ | $48 \times 48$ | |
| **lattice type** | $p_c \pm$ s.d. | $p_c \pm$ s.d. | $p_c \pm$ s.d. | $p_c \pm$ s.d., (lattice size) |
| square (s) | $0.507 \pm 0.090$ | $0.582 \pm 0.032$ | $0.580 \pm 0.018$ | 0.569, ($78 \times 78$) |
| triangular (s) | $0.435 \pm 0.029$ | $0.494 \pm 0.037$ | $0.486 \pm 0.017$ | 0.486, ($84 \times 84$) |
| square ($1^{st}$ and $2^{nd}$) (s) | $0.322 \pm 0.047$ | $0.381 \pm 0.029$ | $0.387 \pm 0.014$ | 0.401, ($90 \times 90$) |
| honeycomb (s) | $0.641 \pm 0.061$ | $0.675 \pm 0.027$ | $0.688 \pm 0.015$ | 0.679, ($\frac{3}{4} \times 84 \times 84$) |
| kagome (s) | $0.609 \pm 0.047$ | $0.643 \pm 0.028$ | $0.635 \pm 0.020$ | 0.655, ($72 \times 72$) |
| four-eight (s) | $0.679 \pm 0.039$ | $0.718 \pm 0.023$ | $0.732 \pm 0.015$ | 0.739, ($84 \times 84$) |
| square (b) | $0.468 \pm 0.049$ | $0.469 \pm 0.028$ | $0.492 \pm 0.011$ | 0.498, ($\frac{3}{4} \times 96 \times 96$) |
| triangular (b) | $0.279 \pm 0.038$ | $0.324 \pm 0.046$ | $0.329 \pm 0.021$ | 0.349 |
| kagome (b) | $0.419 \pm 0.058$ | $0.432 \pm 0.045$ | $0.449 \pm 0.032$ | 0.435, ($84 \times 84$) |
| four-eight (b) | $0.615 \pm 0.050$ | $0.649 \pm 0.028$ | $0.675 \pm 0.027$ | 0.661, ($\frac{3}{4} \times 78 \times 78$) |

**Table 4.5** *Critical probabilities and the standard deviation (s.d.), $p_c \pm$ s.d., for sites (s) and bonds (b), from Dean (1963).*

The effect of shape of arrays found by him is such that for array sizes of $6 \times 96$, $12 \times 48$, and $24 \times 24$ the percolation probabilities are respectively $0.707 \pm 0.043$, $0.594 \pm 0.039$, and $0.568 \pm 0.032$. The *modified second moment* of the cluster size distribution for a lattice has been defined as $\mu_2^* = \sum \sigma_i^2 / (\sum \sigma_i)^2$, where $\sigma_i$ is the size of the $i^{th}$-cluster. And the percolation probability for the finite lattice has been defined to be the value of $p$ at which $\Delta\mu_2^*/\Delta p$ is maximum, $\Delta$ being an increment of one step.

Tiyapan (1995) finds from 27 simulations on 2-d Voronoi networks of between 70 and 500 cells the percolation probability of cells $p_c = 0.507$, and from 75 simulations on 2-d Voronoi networks of between 100 and 8,600 bonds $p_c = 0.658$. For honey comb lattices between 100 and 3,000 bonds he finds the bond percolation probability $p_c = 0.640$, for Kagome lattices between 300 and 3,000 bonds $p_c = 0.517$, for square lattices between 200 and 2,000 bonds $p_c = 0.467$, and for triangular lattices

between 300 and 3,000 bonds $p_c = 0.341$. These results from simulations are usually less than the exact values by a few per cent.

Mecke and Seyfried (2002) find pseudocritical threshold for different types of lattices and then extrapolate these by using finite-size scaling laws to obtain the critical probability for the infinite network. They use the Hoshen-Kopelman method to find the largest cluster of the system and then determine the percolation in fifteen steps by increasing or decreasing $p$ by $\frac{\Delta p}{2}$ depending on whether the system percolates or not. Here $\Delta p = p_2 - p_1$, $p_2 > p_1$, such that the system only percolates at $p_2$ not $p_1$. In two dimensions the network percolates when two opposite sides connect with each other whereas in three dimensions the same is true when four opposite sides connect together. The pseudocritical percolation threshold is $p_c(L)$, where $L$ is the network size. The finite-size scaling theorem is $p_c(L) = p_c(\infty) + aL^b$. In two dimensions they find $p_c$ =0.59278(4), while in three dimensions $p_c = 0.31162(8)$.

In its early days the percolation theory concerned itself much with self-avoiding walks. The connective constant or the walk limit is defined as a measure of the connectivity of the lattice as $\ln \mu = \lim_{n\to\infty}(1/n) \ln c_n$, or sometimes as $\kappa = \ln \mu$. Shante and Kirkpatrick (1971) gives results from both the Monte Carlo and series method as shown in Table 4.6.

| | $z$ | $\mu$ | **Monte** | **Carlo** | **series** | **method** |
|---|---|---|---|---|---|---|
| | | | $p_e$ | $p_v$ | $p_e$ | $p_v$ |
| honeycomb | 3 | 1.8484 | 0.640 | 0.688, 0.679 | 0.6527 | 0.700 |
| kagome | 4 | | 0.435 | 0.655 | | 0.6527 |
| square | 4 | 2.6390 | 0.493, 0.489 | 0.581, 0.569 | 0.5000 | 0.590 |
| triangular | 6 | 4.1515 | 0.341, 0.349 | 0.493, 0.486 | 0.3473 | 0.5000 |
| diamond | 4 | 2.878 | 0.390 | 0.436 | 0.388 | 0.425 |
| s.c. | 6 | 4.6826 | 0.254 | 0.325 | 0.247 | 0.307 |
| b.c.c. | 8 | 6.5288 | | | 0.178 | 0.243 |
| f.c.c. | 12 | 10.0350 | 0.125 | 0.199 | 0.119 | 0.195 |
| h.c.p. | 12 | | 0.124 | 0.204 | | |

**Table 4.6** *Critical probabilities and connective constants, Shante and Kirkpatrick (1971), where s.c. means simple cubic, b.c.c. body centred cubic, f.c.c. face centred cubic and h.c.p. hexagonal close packing.For critical probabilities, all the values given in four decimal places are exact where as those given in three decimal places are numerical.*

## § 4.7 Percolation of $n$-gons in continuum

Consider the continuum percolation of regular polygons in two dimensions. Polygons are placed on the plane randomly with regards to both their position and orientation. From their position one can find the Delaunay triangulation. Using the bonds of the triangulation obtained as reference axes, one for each pair of polygons, one then finds the orientation of any two neighbouring polygons with reference with the axis connecting them.

Unlike in the case of Voronoi percolation where the size of each cell depends on the density of the generating points, in continuum percolation the size of the polygons has been decided in advance. The expanse of the region of interest is then determined relative to this size. The area being simulated can then be imagined as lying within an infinite plane where the number density of polygons is homogeneous.

The area of the n-gon is $A = n \cdot 2 \cdot \frac{1}{2} yz$. But $y = R \sin\theta$ and $z = R\cos\theta$, where $R$ is the radius of the circumscribing circle. Therefore we have $R = \frac{A}{n \sin\theta \cos\theta}$. The radius of the inscribed circle is $r = z$.

A list is made of every angle that the rays perpendicular to the sides of the polygons make with the $x$-axis in the counterclockwise direction. Then the key procedure of the algorithm is to find for each pair the ray which lies closest to the reference line joining the two polygons. The angles that these two rays make with the reference axis determines whether or not the two polygons intersect.

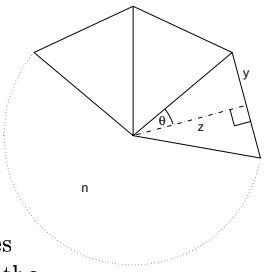


Figure 4.5

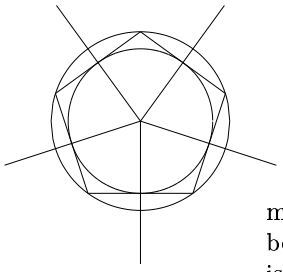


Figure 4.6

The inscribed circle and the circumscribing circle both play a role in deciding whether two polygons form a cluster. Let the radius of the former be represented by $r$ and the radius of the latter $R$. If any two polygons are less than $2r$ apart, then they must overlap. They just touch if they are exactly $2r$ apart. And if the distance between them is more than $2R$, then they can never touch each other. When the distance lies between $2r$ and $2R$ there is a probability that they will touch. Whether this is the case or not depends on the orientation of the two polygons.

Among all the rays similar to those shone in Figure 4.7 one can find the one closest to the line between the centres of the two polygons, ie $\theta_1$ and $\theta_2$ of Figure 4.8 are a pair of minimum angles for $c_1$ and $c_2$. Let the centres of these polygons be $c_1$ and $c_2$, and the line joining them $\overline{c_1c_2}$. Two such lines obtained from the two polygons can fall into on of the two cases; they can either be on the same side of $\overline{c_1c_2}$, or they are on the opposite side of each other as shown in Figure 4.8 (a) and (b) respectively. In either case, the following theorem is true.

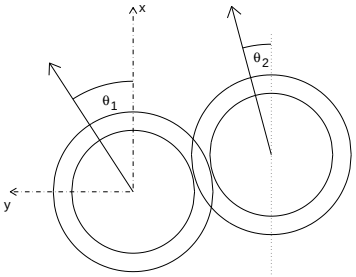


Figure 4.7

**Theorem 4.1.** *The line segments $\overline{a_1a_2}$ and $\overline{b_1b_2}$ always intersect $\overline{c_1c_2}$.*

**Proof.:** Without losing generality, consider only the case of $\overline{a_1a_2}$. Suppose the above statement is false and $\overline{a_1a_2}$ did not intersect $\overline{c_1c_2}$. Then, because $\overline{a_1a_2}$ is a part of the circumference of a polygon, there would always be another ray $d_k$ originating from $c_1$ which has $\overline{a_ia_j}$ connected to it, such that $\overline{a_ia_j}$ intersects $\overline{c_1c_2}$, $i, j$ and $k$ being positive intergers. But the two triangles $\triangle c_1a_1a_2$ and $\triangle c_1a_ia_j$ are identical. Therefore the angle $\theta_k$ between $\overline{a_ia_j}$ and $\overline{c_1c_2}$ would be smaller than $\theta_1$ and we replace $d_1$ with $d_k$ because the latter is closer to $\overline{c_1c_2}$. □

Furthermore, between $\theta_1$ and $\theta_2$ the one which is greater would belong to an *aggressive* cell, while the other one is in a *passive* state. Suppose that $\theta_1 > \theta_2$, then $c_1$ would be the one which does the touching first. Theorem 4.2 states this in a more formal manner.

**Theorem 4.2.** *If the distance between any two identical regular polygons allows them a certain probability $p$ of overlap, such that $0 \leq p \leq 1$, then the polygon whose ray is furthest from the line connecting their centres will be the one which causes the two to intersect.*

**Proof.:** Let $c_1$ and $c_2$ be the two identical polygons in question. As mentioned earlier, there are two cases to be considered, which are represented by Figure 4.8 (a) and (b). Because the distance

between them is more than the diameter of the inscribed circle, when $\theta_1 = \theta_2 = 0$ the two polygons do not touch each other. The two polygons being identical, it is suffice to consider only one of them. But because there is an overlap area between the two as shown in Figure 4.7 where there is a probability of colliding to occur, and as the distance from a centre $c_1$ increases as one goes from a midpoint $d_1$ of an edge to the next vertex $d_2$ closest to $\overline{c_1c_2}$, the greater $\theta_1$ is, the further away from $c_1$ is the point of intersection between $\overline{d_1b_2}$ and $\overline{c_1c_2}$, the more that polygon overlaps into the collision hazard zone mentioned above, and therefore the more contribution to the probability of the collision. □

**Alternative proof.** As shown in Figure 4.8 (c), with $\theta_1 > \theta_2$ it is the corner $a_2$ which collides with the flat of the side $b_1b_2$ and penetrates into the second polygon. Imagine the polygons revolving around until they collide. At the point of collision if their vertices touch, then $\theta_1 = \theta_2$ and it is a special case where both contribute equally to the touch. The case of Figure 4.8 (b) where the two angles lie on the opposite side of each other follows a similar line of reasoning. □

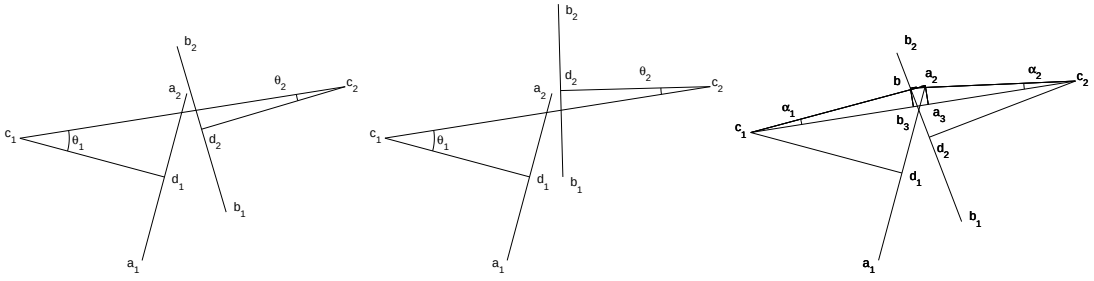



**Figure 4.8** *The distance between $c_1$ and $c_2$ is less than the diameter of the circumscribing circle but more than the diameter of the inscribed circle.*

If $\theta_1 > \theta_2$, then $d(\overline{c_1a_3}) > d(\overline{c_2b_3})$. Furthermore, a touch or penetration implies that $d(\overline{c_1a_3}) + d(\overline{c_2b}) = d(\overline{c_1c_2})$. If $d(\overline{c_1a_3}) + d(\overline{c_2b}) < d(\overline{c_1c_2})$ then the two polygons do not touch each other. The case when they merely touch at vertices is the case where $d(\overline{c_2b}) = d(\overline{c_2b_3})$. On the other hand, the condition $d(\overline{c_1a_3}) + d(\overline{c_2b_3}) \geq d(\overline{c_1c_2})$ does not say anything much because $a_2$ may just pass under $b_2$ without touching the side in Figure 4.8 (a), or it may be the case where the two opposing sides lean towards the same direction as in Figure 4.8 (b). Therefore, we now have the following theorem which is later used in Algorithm 4.6 for finding overlaps.

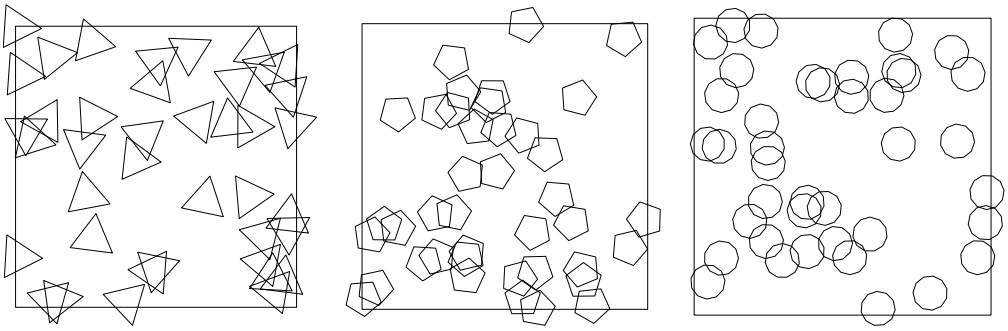
**Theorem 4.3.** *Any two polygons $p_1$ and $p_2$ touch each other if and only if $d(\overline{c_1a_3}) + d(\overline{c_2b}) = d(\overline{c_1c_2})$.*

**Proof.:** See explanation above together with Figure 4.8. □

**Algorithm 4.6** *Find continuum percolation of regular polygons on a plane.*

**place** polygons randomly;
**find** radii of their inscribed circles and the circumscribing circles;
**find** coordinates of the vertices;
**find** Delaunay triangulation;
**for** $i = 1$ **to** *number of triangles* **do**
  **for** $j = 1$ **to** 3 **do**
    **find** *distance between the $j^{th}$-pair vertices of the $i^{th}$ triangle;*
    **if** this *distance* $\leq 2r$ **then** polygons overlap **do**
      **record** cells overlap;
    **elseif** *distance* $\leq 2R$ **then** polygons could overlap **do**
      **record** cells to find whether overlap;
    **endif**
  **endfor**
  **find** *angle* of each of the two polygons when viewed from the other
**endfor**
**for** *every side of every polygon* **do**
  **find** *angles of the normal ray counterclockwise from the positive x-axis;*
**endfor**
**for** *every cell which could still overlap* **do**

**find** *normal rays closest to the lines going towards its neighbours;*
**endfor**
**complete** the overlap check for all remaining cells;
**check** percolation; □



(a) (b) (c)

Figure 4.9 **(a)** Randomly placed *triangles.* Here the program tells us that the largest cluster has seven triangles, and the next largest one has six. There are 8 isolated triangles, 5 clusters of two, and three clusters of three. According to the program, the two clusters of sizes six and two on the upper right corner do not quite touch. **(b)** The largest cluster in this case has got nine *pentagons.* **(c)** These 11-*gons* are also too sparsely placed to percolate. The largest cluster in this case has got 8 members. The number of all 11-gons is 40, the same as in the two previous cases, and six of which are isolated. There are one cluster of four, four clusters of three, and five clusters of two.

The program in § A.5 uses the theorems above to find whether an aggregate of polygons percolate or not. Here each polygon has a unit area. The area of an $n$-gon is $nr^2\cos(\theta/2)\sin(\theta/2)$, where $r$ is the radius of its circumscribed circle. By setting this area to one we can find $r$, which, together with the decided orientation of each polygon gives the coordinates of its vertices. The program is explained in Algorithm 4.6.

A percolation threshold normally means the critical percentage by number. But in the case of percolation in a continuum where there is no fixed amount of lattice sites to refer to, the per cent area covered may be a better candidate for $p_c$. Finding this area can become quite computationally intensive it is a problem of finding the union among sets which, for the case of three sets is $A \cup B \cup C = A+B+C-AB-AC+ABC$, for four sets $A \cup B \cup C \cup D = A+B+C+D-AB-AC-AD-BC-BD-CD+ABC+ABD+ACD+BCD-ABCD$, *etc.* For the general case where there are $n$ sets intersecting one another, then, $\bigcup_n A_i = \sum_n Ai - \sum_{(n,2)} A_{ij} + \sum_{(n,3)} A_{ijk} - \cdots + \sum_n A_{i^{(n-1)}} + (-1)^{(n+1)} A_{i^{(n)}} = \sum_k^n (-1)^{(k+1)} A_{i^{(n)}}$, where $A_{ijk\cdots}$ means $A_i \cap A_j \cap A_k \cap \cdots$, the subscripts $(n,k)$ is the combination ${}^n\mathrm{C}_k = n!/\left[k!(n-k)!\right]$ and $i^{(n)} = ijk\cdots$ up to the $n^{\text{th}}$ term.

## § 4.8 Polygon percolation threshold

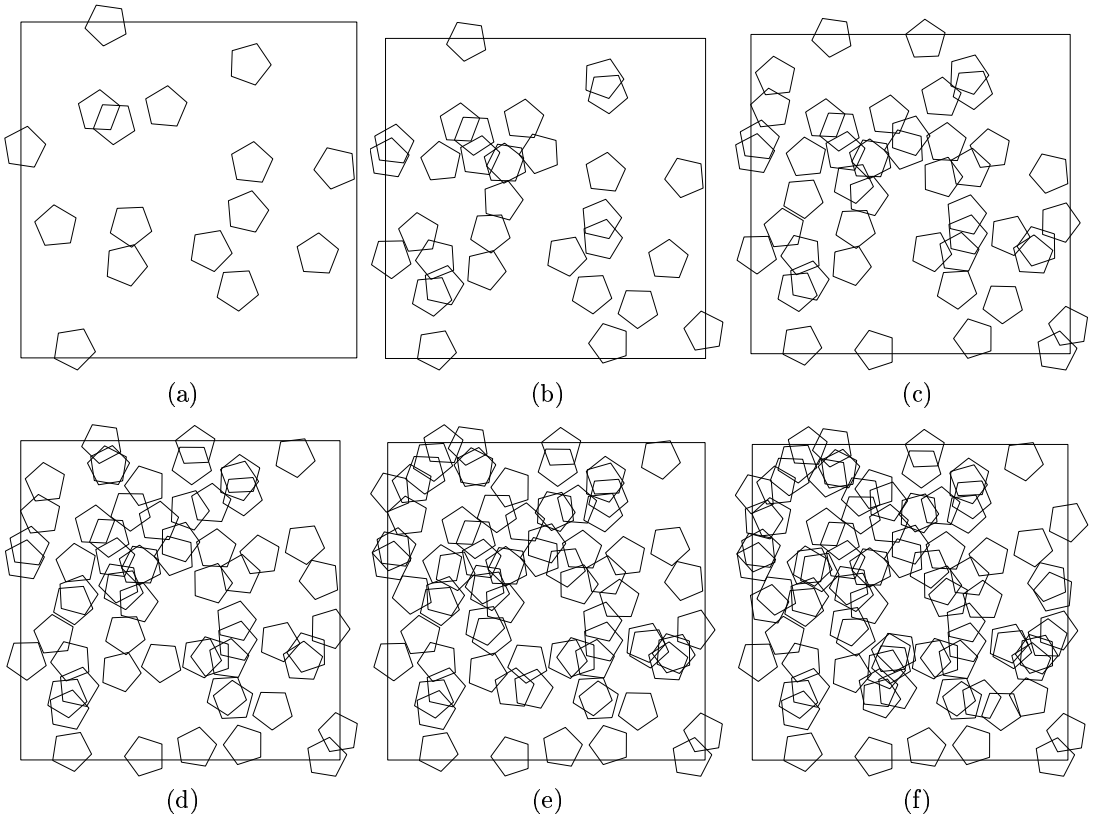
Because, unlike lattices however random, the continuum has neither underlying sites nor vertices and therefore has no definite number of these. To describe the percolation threshold in such situation we need to define the some new parameter other than our usual $p_c$, which not only depends on the number of polygons but also their area relative to that of the domain which they are in. We define the threshold area ratio as Definition 4.1. This definition disregards the intersection of the polygons, and therefore can only be used to compare networks which have the same type of distribution.

**Definition 4.3.** *For an aggregate of n-gons all the centres of which are Poisson points within a space of area A, the threshold area ratio is* $\alpha_n = \sum_n |a_i|/A$, *where* $a_i$ *is the area of each n-gon. If all n-gons have an equal area, then* $\alpha_n = na/A$.:

The program in § A.5 is transformed into a function which is then reused several times in the course of operation of another program, listed in § ℵ, which finds $\alpha_n$. Both the program, `ppgt.m`, and the function, `cmpc.m`, are here listed concatenated together whereas in practice they must separately exist.

The program first proceeds to find whether percolation occurs for aggregates containing increasing numbers of pentagons in step of 16. This step size is in fact $2^k$ for some integral $k$, and can be chosen to be optimum for a certain size of the network. At the end of these steps we find whether our aggregate percolate. If it does, then we know that somewhere within the last increasing step exists the point where percolation sets in. The precise location of this point can then be found by doing binary searching. There are altogether $k$ steps of these searches, at the end of each one of which there is again a test for percolation. If this test for the last and $k^{\text{th}}$ step is successful, then the last picture represents the aggregate which starts to percolate, otherwise one needs to know how many consecutive percolation tests have failed up to that point. With the knowledge of that number, one can count backward and reach the picture in question. The program `ppgt.m` fails to do this last job, but it has been added, together with the generalisation to $2^k$ steps, to make `ppgk.m`.

Figure 4.10 is the sample run on an aggregate of pentagons. Each of the pentagons is one unit area, and the space containing them, or rather the space which contains all their centres, is $10 \times 10$ in size.



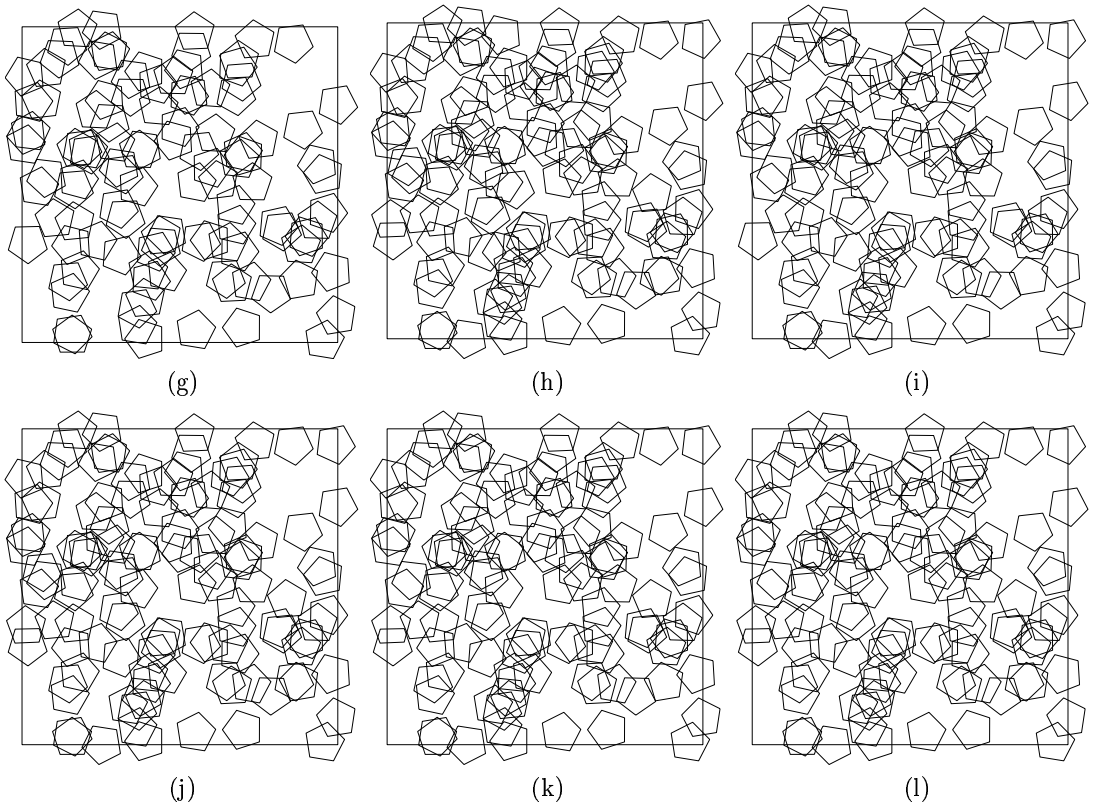



**Figure 4.10** *Finding the threshold area ratio of pantagons. The number of pentagons from* (a) *to* (h) *increases in step of* $2^4 = 16$. *In* (i), (j), (k) *and* (l) *a binary search proceeds which divides the length of the last interval from 16 to 8, 4, 2 and 1 in sequence. This simulation gives* $\alpha_5 = 1.23$.

The program `ppgk.m` in § ℵ uses $k = 4$ as explained above. Generalising this into a general $k$ gives Algorithm 4.7. Here $v_{ij}$ is the $j^{\text{th}}$ vertex of the $i^{\text{th}}$ $n$-gon. At the completion of the program, if it does complete, we have the aggregate containing $m_{1/2}$ polygons, and our $b^{\text{th}}$ percolation test before last has been on this aggregate.

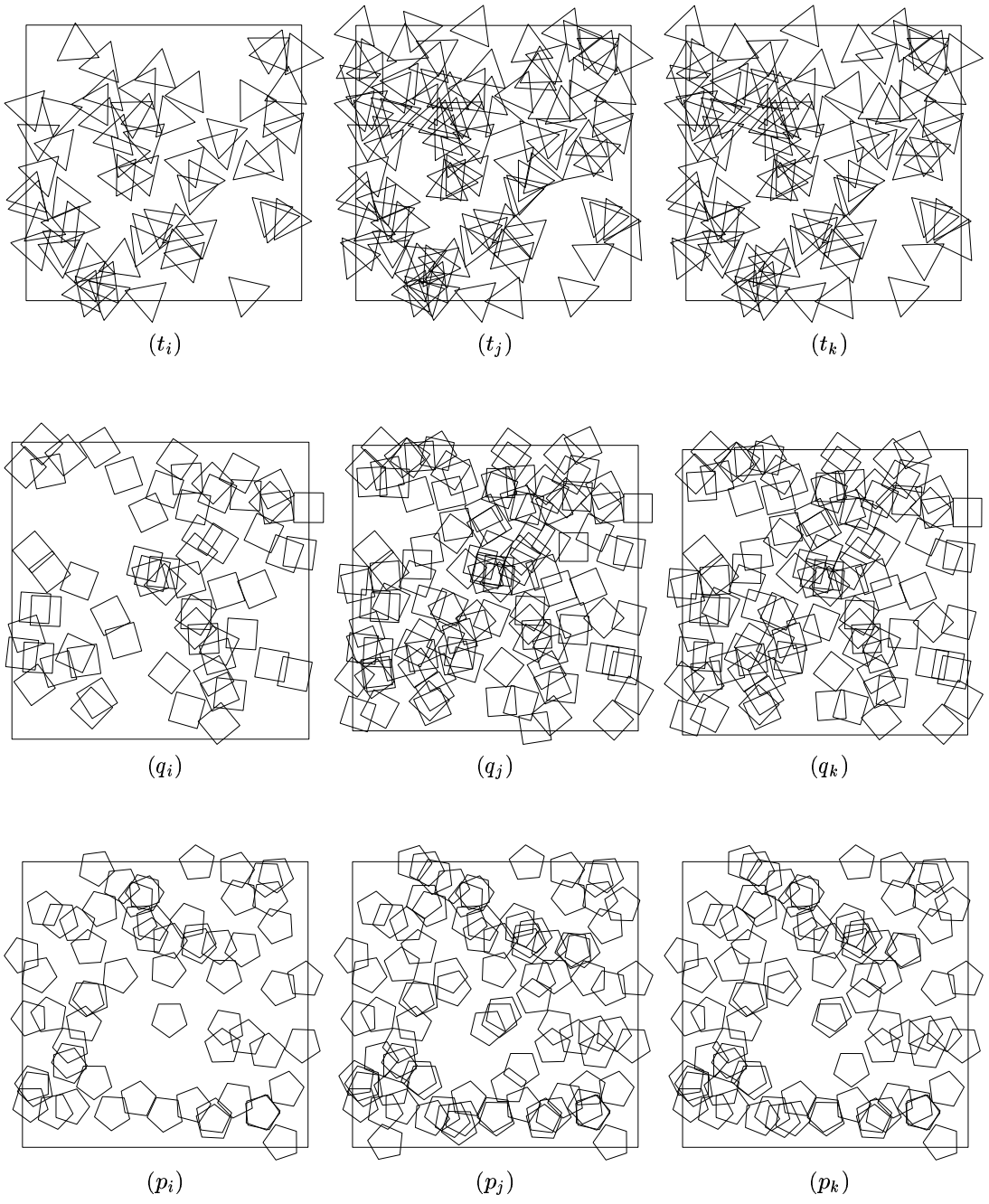
**Algorithm 4.7** *Threshold area ratio.*

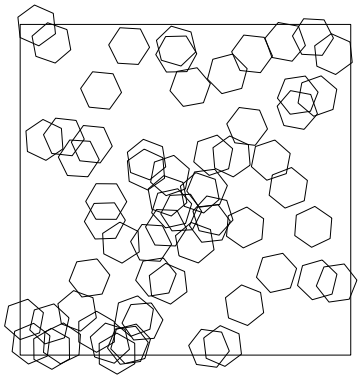
$s \leftarrow 2^k$;
$\beta \leftarrow 2\pi/n$;
$r \leftarrow [1/(n \sin(\beta/2)\cos(\beta/2))]^{1/2}$;
**while** percolation check fails **do**
  $\{(x, y)\} \leftarrow$ **find** $s$ more centre coordinates;
  $\{\theta\} \leftarrow$ **find** $s$ more orientational angles;
  **for** all these new centres and angles **do**
    $v_{ij} \leftarrow (x + r\cos(\theta_i + j\beta), y + r\sin(\theta_i + j\beta))$;
    **check** percolation all the $m$ centres and angles;
  **endfor**
**endwhile**
$(m_0, m_1] \leftarrow$ the last step above;
$b \leftarrow 0$;
**repeat** $k$ times
  $m_{1/2} \leftarrow (n_0 + n_1)/2$;
  **check** percolation all the first $m_{1/2}$ angles and centres;
  **if** percolated **then**
    $m_1 \leftarrow m_{1/2}$;
    $b \leftarrow 0$;
  **else**
    $m_0 \leftarrow m_{1/2}$;
    $b \leftarrow b + 1$;

**endif**

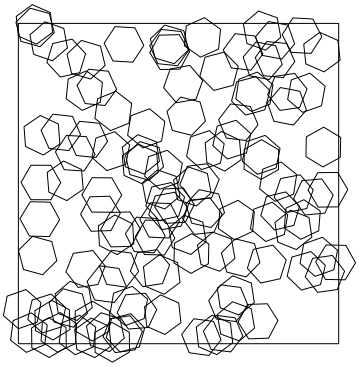
**endrepeat** □

Figure 4.11 shows the percolation of $n$-gons with various values of $n$, the results of applying Algorithm 4.7 above. Polygons up to the twelve-sided dodecagon are shown here. After this there are of course the triskaidecagon, tetrakaidecagon, ..., enneakaidecagon, icosagon, icosikaihenagon, icosikaidigon, icosikaitrigon, ..., icosiaienneagon, triacontagon, triacontakaihenagon, ..., triacontakaienneagon, tetracontagon, ..., pentacontagon, ..., hexacontagon, ..., heptacontagon, ..., octacontagon, ..., enneacontagon, ..., hectagon, *etc.* But all of these closely approaches the circle.

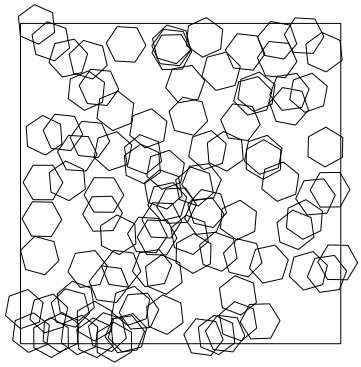




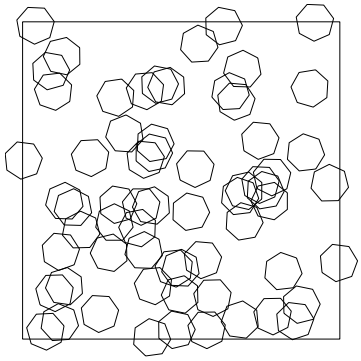
$(h_i)$



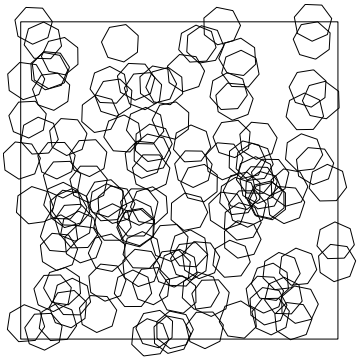
$(h_j)$



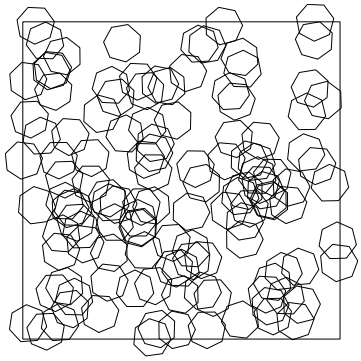
$(h_k)$



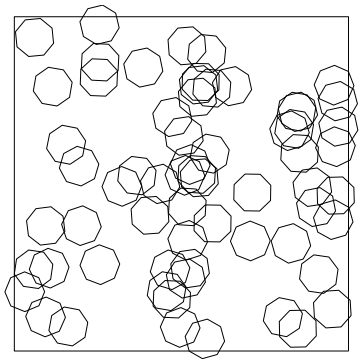
$(k_i)$



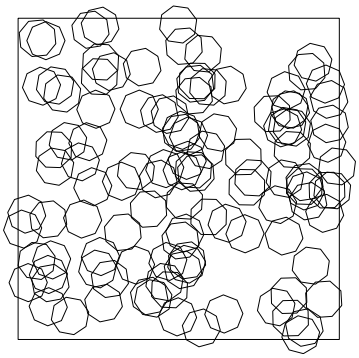
$(k_j)$



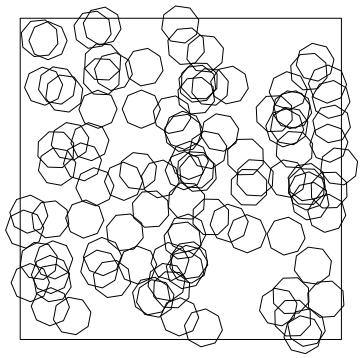
$(k_k)$



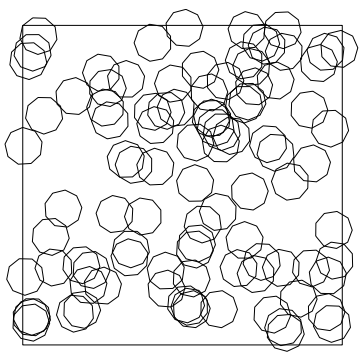
$(o_i)$



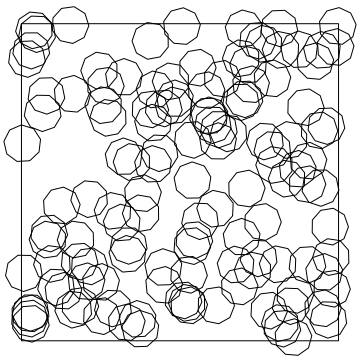
$(o_j)$



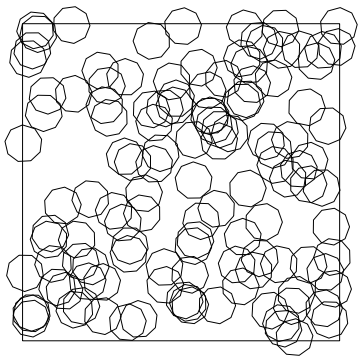
$(o_k)$



$(e_i)$



$(e_j)$



$(e_k)$

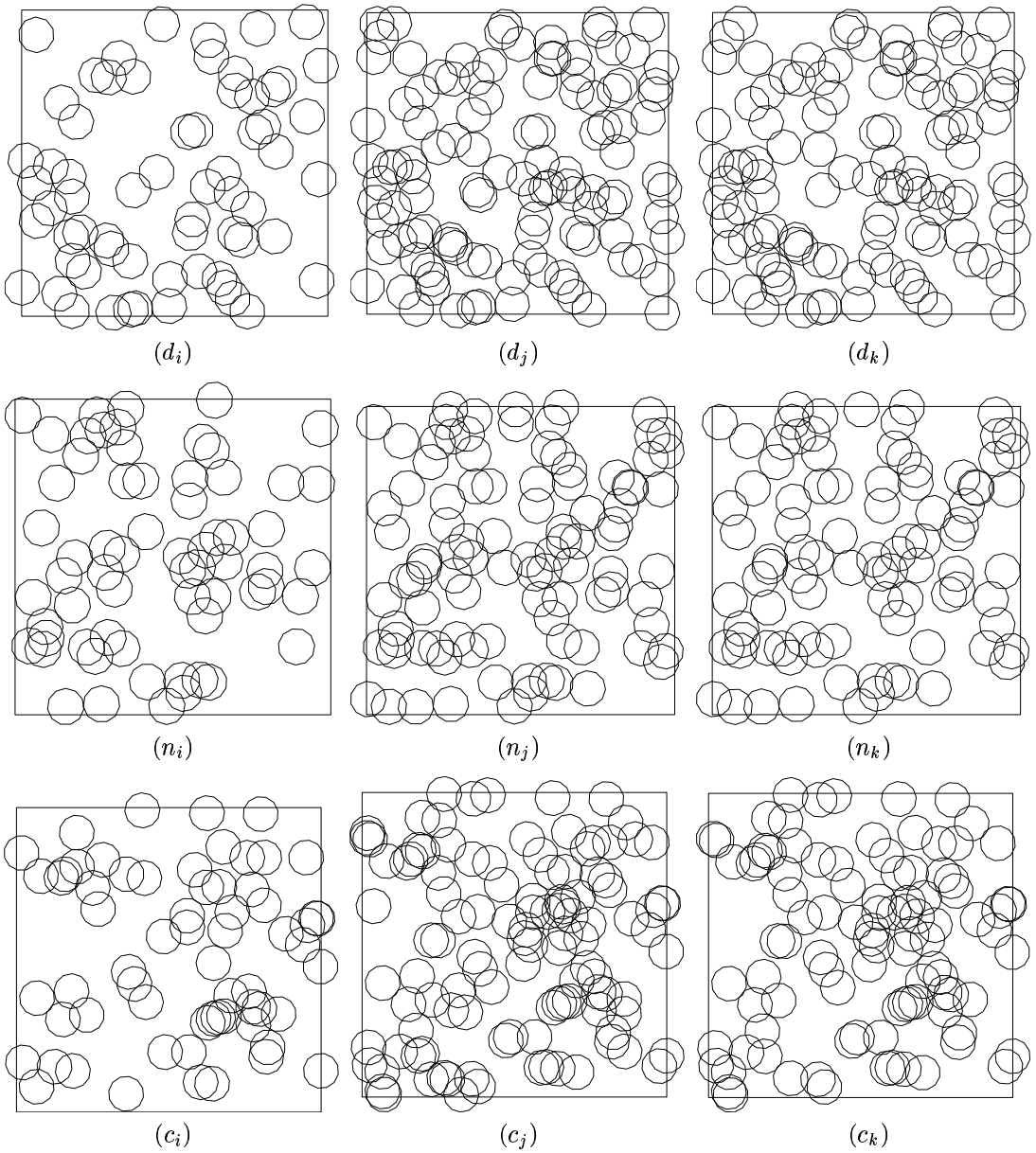
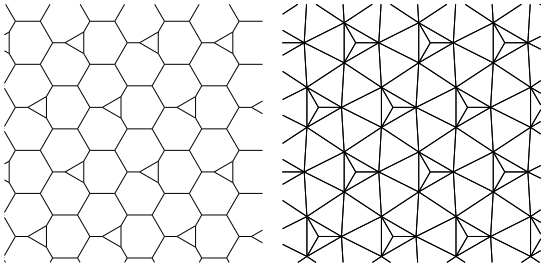


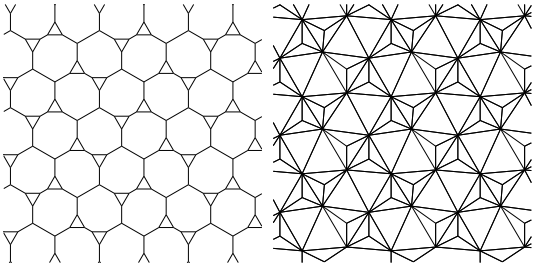

**Figure 4.11** *Percolation of n-gons. Triangle, square, pentagon, hexagon, heptagon, octagon, enneagon, decagon, hendecagon and dodecagon are represented respectively by t, q, p, h, k, o, e, d, n and c. The subscripts i, j and k are respectively the fourth step of the program, the over-percolated case and the critically percolated case.* $\alpha_3 = 1.03$, $\alpha_4 = 1.17$, $\alpha_5 = 0.89$, $\alpha_6 = 0.99$, $\alpha_7 = 1.23$, $\alpha_8 = 1.03$, $\alpha_9 = 1.23$, $\alpha_{10} = 1.07$, $\alpha_{11} = 0.91$ *and* $\alpha_12 = 0.95$.

## § 4.9 2-homohedral tilings

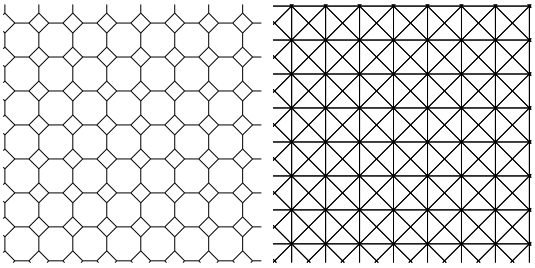
There are altogether 39 types of 2-homohedral tilings, ie. those which have vertices of the same valence. Of these, there are 26 types of valence 3, 10 of valence 4, and 3 of valence 5 (*cf* Grünbaum *et al*, 1987). The code and the data used in the simulations are in § A.6. In the program, the variable $o$ contains the vertex numbers which must be ordered by scanning the basic tile from left to right and gradually from bottom up. Unless the variable follows this ordering strange results will occur. The variables $m$ and $n$ are respectively the $x$- and $y$-coordinates of the vertices.



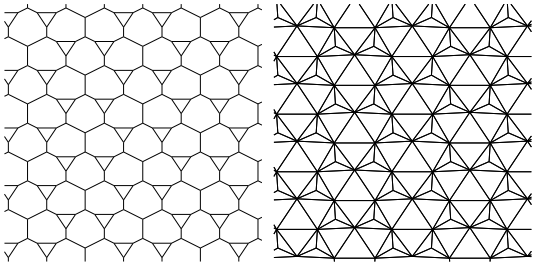
**(1)** $3_3[3^3]7_1[3^7]$



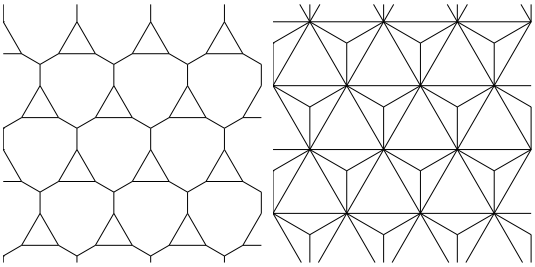
**(2)** $3_3[3^3]9_3[3^9]_{\rm II}$



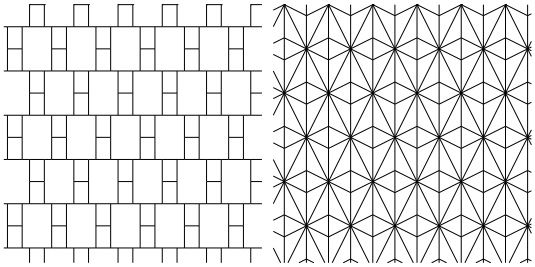
**(3)** $4_4[3^4]8_4[3^8]$



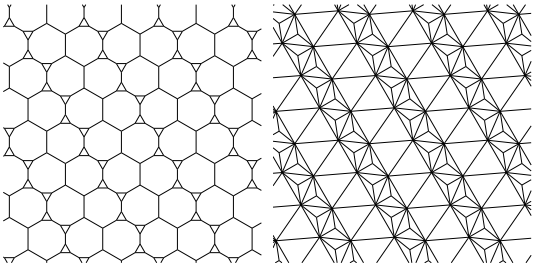
**(4)** $3_3[3^3]8_2[3^8]$



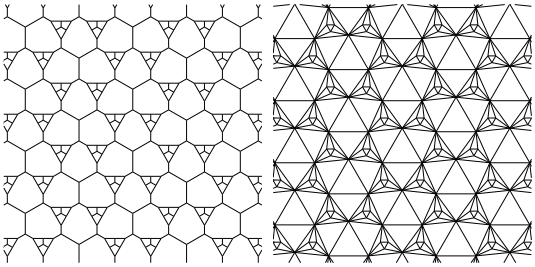
**(5)** $3_3[3^3]9_3[3^9]_{\rm III}$



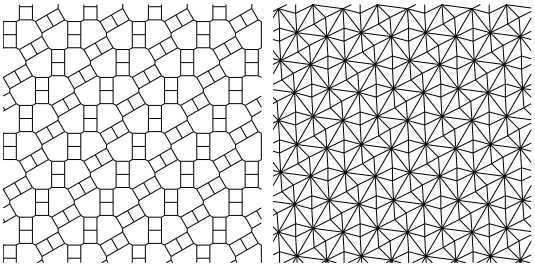
**(6)** $4_3[3^4]10_6[3^{10}]_{\rm I}$



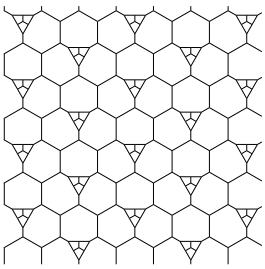
**(7)** $3_3[3^3]9_3[3^9]_{\rm I}$

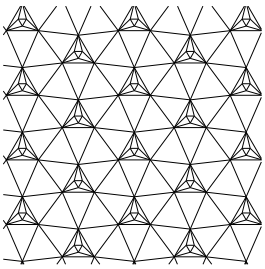


**(8)** $4_2[3^4]10_4[3^{10}]$

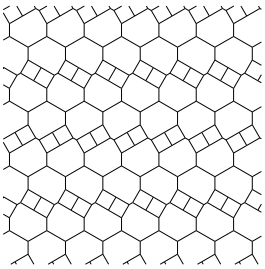


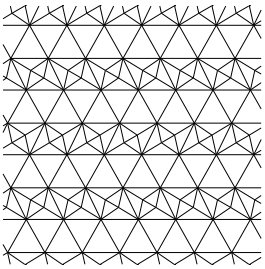
**(9)** $4_3[3^4]10_6[3^{10}]_{\rm II}$



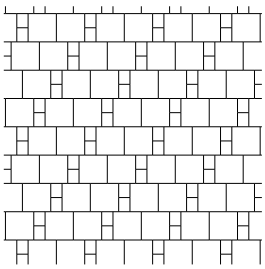


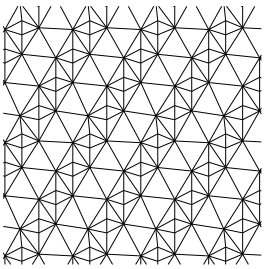
**(10)** $4_2[3^4]8_2[3^8]$



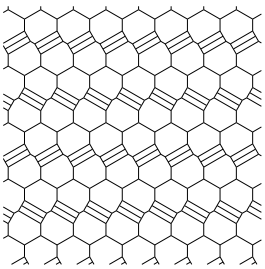


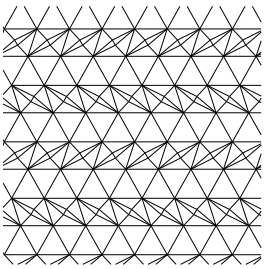
**(11)** $4_3[3^4]8_3[3^8]_{\mathrm{I}}$



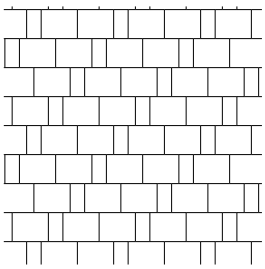


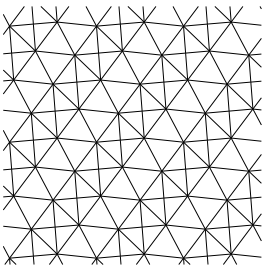
**(12)** $4_3[3^4]8_3[3^8]_{\mathrm{II}}$



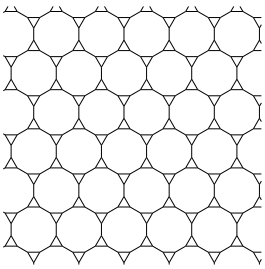


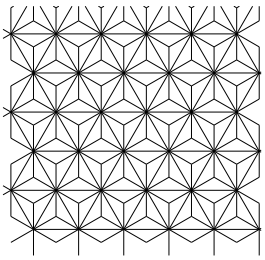
**(13)** $4_3[3^4]8_3[3^8]_{\mathrm{III}}$



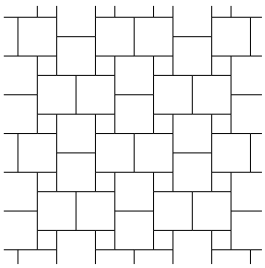


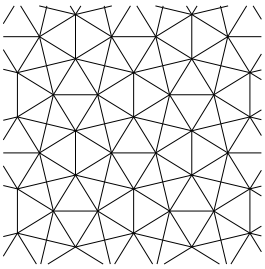
**(14)** $4_4[3^4]7_2[3^7]_{\mathrm{II}}$



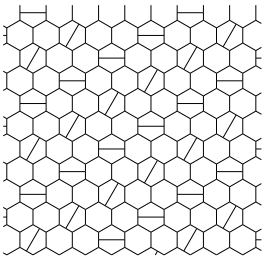


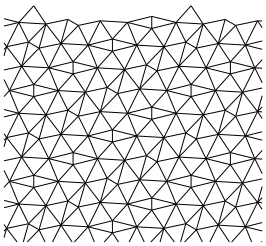
**(15)** $3_3[3^3]12_6[3^{12}]$



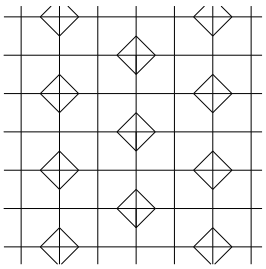


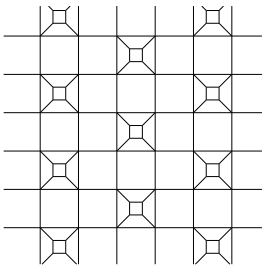
**(16)** $4_4[3^4]7_2[3^7]_{\mathrm{I}}$

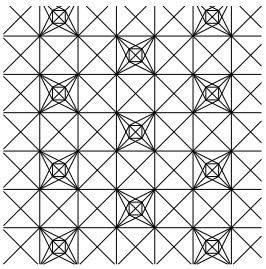




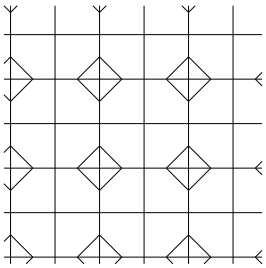
**(17)** $5_4[3^5]7_4[3^7]_{\mathrm{I}}$

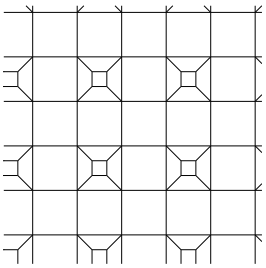


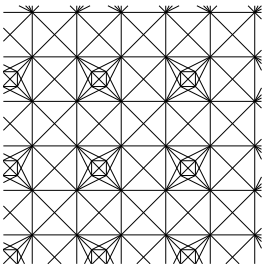




**(18)** $3_1[4^3]5_1[4^5]_{\mathrm{I}}$







**(19)** $3_1[4^3]5_1[4^5]_{\mathrm{II}}$